EDUARDO DA SILVA

GERENCIAMENTO DE CHAVES PÚBLICAS SOBREVIVENTE BASEADO EM GRUPOS PARA MANETS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Carlos Pessoa Albini
Coorientador: Prof. Dr. Aldri Luiz dos Santos

CURITIBA

2009

EDUARDO DA SILVA

GERENCIAMENTO DE CHAVES PÚBLICAS SOBREVIVENTE BASEADO EM GRUPOS PARA MANETS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Carlos Pessoa Albini
Coorientador: Prof. Dr. Aldri Luiz dos Santos

CURITIBA

2009

EDUARDO DA SILVA

GERENCIAMENTO DE CHAVES PÚBLICAS SOBREVIVENTE BASEADO EM GRUPOS PARA MANETS

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Informática da Universidade Federal do Paraná, pela Comisso formada pelos professores:

Orientador: Prof. Dr. Luiz Carlos Pessoa Albini
Departamento de Informática, UFPR

Prof. Dr. Aldri Luiz dos Santos
Departamento de Informática, UFPR

Prof. Dr. Célio Vinicius Neves de Albuquerque
Instituto de Computação, UFF

Prof. Dr. André Luiz Pires Guedes
Departamento de Informática, UFPR

Curitiba, 09 de julho de 2009

AGRADECIMENTOS

Primeiramente, eu quero agradecer a Deus, por ter permitido que eu alcançasse esse objetivo, dando-me forças e capacidade para trilhar esse caminho. Também quero agradecer a meus familiares, principalmente à minha amada esposa Jesli, que tem sido minha fortaleza em todos os momentos difíceis e meu ânimo para vencer todos os desafios. Ao meu filho Miguel, que chegou para alegrar o nosso lar durante o andamento desse mestrado. Quero extender esse agradecimento à minha mãe, Maria Terezinha, pelo exemplo e dedicação, e aos meus irmãos Marcos, Thiago e Vanessa.

Um agradecimento especial aos meus orientadores, prof. Albini e prof. Aldri. Eles me ensinaram, nesse período, como fazer pesquisa com seriadade e competência. Muitas e muitas vezes tiveram que parar e alinhar os meus pensamentos com aquilo que se espera de um pesquisador. Muito obrigado, e ainda estarei aí por algum tempo, para aprender cada dia mais. Também quero agradecer à Michele, que esteve desde o início desta jornada me auxiliando, ensinando e cobrando. Mesmo estando na França, você foi fundamental nesse processo. Agradeço a todos os membros do grupo NR2: é muito bom fazer parte de um grupo como esse.

Também gostaria de agradecer à minha empresa, SOCIESC, e a todos os meus colegas de trabalho. Ao Marco André e ao Mehran, um muito obrigado por esterem sempre me incentivando, desde o meu estágio em informática (antes de iniciar a graduação). Quero agradecer àqueles que, mesmo sem saberem, auxiliaram indiretamente nesse processo, com suas criações. Nesta lista encontram-se: Donald Knuth (TEX), Leslie Lamport (LATEX), Linus Torvalds (Núcleo do GNU/Linux), Richard Stallman (Projeto GNU), Guido van Rossum (linguagem Python), Larry Page e Sergey Brin (Google), Paul Buchheit (Gmail), Mark Shuttleworth (Ubuntu), e tantos outros desenvolvedores, que com seus projetos facilitam a vida de milhares de pessoas.

Muito obrigado!

Que Deus me conceda falar com inteligência e ter pensamentos dignos dos dons que recebi, pois é Ele quem guia a Sabedoria e dirige os sábios. Nós estamos nas suas mãos, nós e nossos discursos, toda a nossa inteligência e nossa habilidade. Foi ele quem me deu o conhecimento exato de todas as coisas.

Sabedoria 7,15-17a

RESUMO

As características particulares das redes ad hoc móveis, principalmente a topologia dinâmica e a ausência de infraestrutura, dificultam a implementação de sistemas de gerenciamento de chaves eficazes para essas redes. Dentre os diversos sistemas propostos na literatura, o Sistema de Gerenciamento de Chaves Públicas Auto-organizado (PGP-*Like*) tem sido aplicado por ser completamente distribuído, auto-organizável, e não depender de uma autoridade certificadora. Inicialmente, este trabalho quantifica os impactos dos ataques de falta de cooperação e *Sybil* no PGP-*Like*. Os resultados mostram, diferente das suposições encontradas na literatura, que este sistema mantém sua eficácia mesmo diante de 40% de nós egoístas, mas é totalmente vulnerável a ataques *Sybil*. Assumindo esses resultados, é apresentado um esquema de gerenciamento de chaves sobrevivente baseado em grupos, mais resistente aos ataques *Sybil* que o PGP-*Like*. Nesse esquema, chamado de *Survivable Group-based Public Key Management for MANETs* (SG-PKM - Gerenciamento de Chaves Públicas Sobrevivente baseado em Grupos), os nós formam grupos baseados na relações de amizades dos usuários, e emitem certificados mutuamente, entre os membros do grupo. O esquema também prevê que os grupos possam emitir certificados para outros grupos, também baseado na relações de amizades. Para que dois nós sem uma conexão direta possam se autenticar mutuamente, eles formam cadeias de certificados que conectam os grupos a que pertencem. Além disso, o esquema exige que sejam formadas no mínimo duas cadeias de certificados de grupos válidos, aumentando a sua resistência contra ataques *Sybil*. Os resultados mostram que o SG-PKM consegue manter a sua eficiência diantes de ataques de falta de cooperação, mesmo na presença de 40% de nós egoístas, resultado similar ao PGP-*Like*. Mais importante, o SG-PKM consegue mitigar o impacto dos ataques *Sybil*, mantendo a taxa de autenticações não comprometidas acima de 70% para grupos com cinco ou seis membros, mesmo na presença de 40% de nós maliciosos.

ABSTRACT

The characteristics of mobile ad hoc networks, as the dynamic environment and the lack of infrastructure, make it difficult the implementation of effective key management systems. Among the proposed systems, the Self-Organized Public Key Management System for MANETs (PGP-Like) has been well considered, as it is totally distributed, self-organized, and does not rely on any certificate authority. Firstly, this work quantifies the impacts of lack of cooperation and Sybil attacks on PGP-Like. Results show that PGP-Like mantains its effectiveness even in face of 40% of selfish nodes, but it is fully vulnerable to Sybil attacks. Thus, the Survivable Group-based Public Key Management for MANETs (SG-PKM) is presented. It is designed to be more resistant to Sybil attacks than PGP-Like. In SG-PKM, nodes form groups based on users relationship, and issue certificates for each other. SG-PKM also stablishes that groups can issue certificates to other groups. Any two nodes, that do not have a direct connection between them, are able to authenticate themselves through certificate chains binding their groups. Moreover, the scheme requires at least two disjoint certificate chains for authentication, increasing the resistance to Sybil attacks. Results show that SG-PKM mantains its effectiveness in face of lack of cooperation attacks, even under 40% of selfish nodes, similarly to PGP-Like. More important, SG-PKM mitigates the impact of Sybil attacks, supporting the non-compromissing authentication rate above than 70% for groups with five or six members, even in presence of 40% of malicious nodes.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LIST OF ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

ACD Autoridade Certificadora Distribuída

ACO *Authentication Communication Overhead* - Sobrecarga de Comunicação para Autenticação

CE *Certificate Exchange Convergence* - Convergência das Trocas de Certificados

CRL *Certificate Revocation List* - Lista de Revogação de Certificados

DFC *Distributed Coordination Function* - Função de Coordenação Distribuída

FIC *False Identity Confidence* - Confiabilidade em uma Identidade Falsa

GR *Group Reachability* - Alcançabilidade dos Grupos no SG-PKM

IA *Indirect Authentication* - Autenticação Indireta

ICP Infraestrutura de Chaves Públicas

IEEE *Institute of Electrical and Electronics Engineers* - Instituto dos Engenheiros Elétricos e Eletrônicos

MAC *Message Authentication Code* - Código de Autenticação de Mensagem

MANET *Mobile Ad Hoc Network* - Rede Ad Hoc Móvel

NCA *Non-Compromised Authentication* - Autenticações Não-Comprometidas no SG-PKM

NCG *Non-Compromised Groups* - Grupos Não-Comprometidos no SG-PKM

PGP *Pretty Good Privacy* - Privacidade Bastante Boa

RCO *Revocation Communication Overhead* - Sobrecarga de Comunicação de Revogação

SAMNAR *Survivable Ad hoc and Mesh Network ARchitecture* - Arquitetura Sobrevivente para Redes Ad hoc e Mesh

SC *Suspicious Certificates* - Certificados Suspeitos

SG-PKM *Survivable Group-based Public Key Management for MANETs* - Gerenciamento de Chaves Públicas Sobrevivente baseado em Grupos para MANETs

SPOF *Single Point of Failure* - Ponto Único de Falhas

TACO *Total Authentication Communication Overhead* - Sobrecarga de Comunicação Total para Autenticação

TTP *Third Trusted Party* - Terceira Entidade Confiável

UA *User Authenticability* - Autenticabilidade do Usuário no SG-PKM

UCO *Update Communication Overhead* - Sobrecarga de Comunicação de Atualização

UR *User Reachability* - Alcançabilidade do Usuário no PGP-*Like*

UR-Req *Update Repository Request Message* - Mensagem de Solicitação de Repositório Atualizado

UR-Rep *Update Repository Reply Message* - Mensagem de Resposta de Repositório Atualizado

VCO *Validation Communication Overhead* - Sobrecarga de Comunicação de Validação

VREP *Validation Reply* - Resposta de Validação

VREQ *Validation Request* - Pedido de Validação

NOTAÇÃO

x_u	identidade de um dado nó u
pk_u	chave pública do nó x_u
sk_u	chave privada do nó x_u
IG	conjunto de grupos do sistema
IG_α	identidade de um dado grupo α
PK_α	chave pública do grupo IG_α
SK_α	chave privada do grupo IG_α
$x_u \rightsquigarrow x_v$	nó x_u é capaz de autenticar o nó x_v
$pk_u \rightarrow pk_v$	certificado do nó v assinado com a chave privada do nó u
$pk_u \rightsquigarrow pk_v$	cadeia de certificados entre os vértices pk_i e pk_v
$PK_\alpha \rightsquigarrow PK_\beta$	cadeia de certificados entre os vértices PK_α e PK_β
$PK_\alpha \rightrightarrows PK_\beta$	cadeia de certificados disjuntos entre os vértices PK_α e PK_β
$C_{SK_\alpha}^{x_i}$	certificado do nó x_i assinado com a chave privada do grupo IG_α
$C_{SK_\alpha}^{IG_\beta}$	certificado do grupo IG_β assinado com a chave privada do grupo IG_α
G_u	repositório local de certificados atualizados do nó u
G_u^N	repositório local de certificados não atualizados do nó u
G	grafo de certificados do sistema
S	conjunto de nós do sistema
NC	conjunto de nós não-comprometidos no sistema
F	conjunto de nós Sybil no sistema
n	quantidade de nós no sistema
$N(x_v)$	conjunto de vizinhos do nó x_v
$\|Z\|$	tamanho de um dado conjunto Z
$a\|\|b$	valor a concatenado com b
$\mathcal{H}(Z)$	função *hash* $\mathcal{H}$ sobre o valor Z

CAPÍTULO 1

INTRODUÇÃO

As redes ad hoc móveis (*Mobile Ad Hoc Networks* (MANETs)) são formadas por um conjunto de dispositivos móveis (nós) que se comunicam entre si usando um canal de comunicação sem fio. Essas redes são estabelecidas dinamicamente sem depender de uma infraestrutura fixa ou uma administração centralizada, e o seu funcionamento é mantido pelos próprios nós de uma forma auto-organizada [49]. A topologia é dinâmica, pois os nós se movimentam livremente pelo ambiente, e podem entrar e sair da rede a qualquer momento sem notificarem uns aos outros. Alguns exemplos de aplicações das MANETs são resgate de emergência, sensores digitais, comunicação em campos de batalha e compartilhamento de dados durante uma conferência [16].

Um do grandes desafios das MANETs é a segurança. Elas herdam todos os problemas de segurança das redes sem fio convencionais como, por exemplo, a escuta não-autorizada dos dados transmitidos [67], que afeta a confidencialidade da comunicação. Somado aos problemas clássicos da comunicação sem fio, a topologia dinâmica das MANETs facilita a ação de adversários, tornando-as susceptíveis a diversos tipos de ataques [24].

1.1 Contextualização do Problema

As MANETs são altamente vulneráveis a ataques passivos e ativos [24]. Em um ataque passivo, um adversário tenta escutar as informações do sistema, mas não afeta os seus recursos. Já em um ataque ativo, ele tenta alterar os recursos do sistema ou afetar a sua operação [58]. Em outras palavras, um ataque é considerado passivo, quando o atacante não interage com a rede, ficando apenas ouvindo ou capturando os dados que estão sendo trafegados no sistema, e um ataque é considerado ativo quando existe alguma modificação de mensagens ou a criação de dados, afetando o comportamento da rede [44, 3].

A criptografia é a principal técnica utilizada para garantir a segurança das redes,

fornecendo integridade, confidencialidade, autenticidade e irretratabilidade do emissor nas comunicações de dados [59]. As técnicas criptográficas dependem de chaves, que são informações relacionadas aos nós comunicantes, e são usadas em conjunto com os algoritmos criptográficos na realização de operações como cifração, decifração e assinatura digital. A administração segura dessas chaves é conhecida como gerenciamento de chaves, e consiste na geração, armazenamento, distribuição, proteção e revogação das chaves, de modo que elas estejam disponíveis aos nós autênticos da rede [43].

Os sistemas criptográficos, também chamados de criptossistemas, podem ser classificados em simétricos e assimétricos. Nos criptossistemas simétricos as operações de cifração e decifração das mensagens são realizadas utilizando a mesma chave [57]. Esse tipo de sistema também é conhecido como "criptografia de chave secreta". Já nos criptossistemas assimétricos, as operações de cifração e decifração são realizadas usando chaves diferentes, uma chave pública e uma chave privada [23]. Também são conhecidos como "criptografia de chave pública".

Entretanto, os criptossistemas simétricos e assimétricos tradicionais têm se mostrado ineficientes em MANETs [45]. Os criptossistemas simétricos, embora exijam pouco processamento, não são escaláveis e os nós comunicantes precisam compartilhar a chave secreta, ou por um canal seguro pré-estabelecido ou antes da formação da rede. Já os sistemas assimétricos tradicionais, necessitam de uma entidade confiável para realizar o gerenciamento das chaves, públicas e privadas, e dos certificados [16]. Contudo, estabelecer uma entidade confiável nas MANETs é uma tarefa difícil, devido à sua organização descentralizada e à ausência de um modelo de confiança entre os nós [7].

É muito comum, nas redes tradicionais ou até mesmo nas redes *ad hoc* militares, a existência de uma autoridade confiável para o gerenciamento da rede. Esses ambientes são chamados de "ambientes gerenciados" [45]. Por outro lado, a utilização de entidades centrais de autenticação em redes maiores pode elevar a necessidade de recursos para o gerenciamento de chaves, e isso nem sempre é desejável. Nas MANETs, a segurança torna-se um desafio ainda maior, pois a criação desses ambientes gerenciados não é possível. Tais ambientes são chamados abertos, e não é desejável a existência de uma entidade central

para gerenciar essas redes. Assim, a confiabilidade da execução correta das operações fica sujeita aos nós participantes da rede, que podem executar suas funções de forma injusta e/ou incorreta.

Dessa forma, o gerenciamento de chaves nas MANETs deve levar em consideração a topologia dinâmica dessas redes e ser auto-organizado e descentralizado [34, 62]. Diversos esquemas de gerenciamento de chaves para as MANETs são encontrados na literatura. Entre eles, um dos principais esquemas é o *Sistema de Gerenciamento de Chaves Públicas Auto-Organizado* [35, 9], chamado neste trabalho de PGP-*Like*. Ele é baseado nos conceitos do PGP [74], no qual os nós criam seus próprios pares de chaves e emitem certificados de chave públicas aos nós de sua confiança. Cada nó possui repositórios locais de certificados, que periodicamente são trocados com seus vizinhos, estabelecendo assim, cadeias de certificados. No entanto, o PGP-*Like* não considera a existência de ataques de má-conduta, oferecendo apenas um mecanismo de detecção de certificados conflitantes.

1.2 Ataques aos sistemas de gerenciamento de chaves nas MANETs

Diferentes tipos de ataques podem danificar os serviços de gerenciamento de chaves nas MANETs. Alguns desses ataques visam comprometer os princípios de disponibilidade, confidencialidade, integridade, autenticidade e irretratabilidade em um sistema de gerenciamento de chaves públicas. Além disso, um nó adversário pode comprometer um ou mais nós e, como consequência, negar ou denegrir as funções do sistema de gerência de chaves. Dentre os principais ataques que podem ser realizados em uma MANET, dois deles podem ter uma grande efeito se aplicados em um sistema de gerenciamento de chaves, sendo eles [24]:

a. *Falta de cooperação:* um nó malcomportado, geralmente chamado de "egoísta", que deseja economizar recursos de energia e processamento pode comprometer o funcionamento da rede simplesmente por não participar de suas operações [44]. A maioria dos sistemas de gerenciamento de chaves para MANETs necessita da

cooperação dos nós para realizarem suas operações. Essa característica é ideal para o fornecimento de serviços em uma rede dinâmica. Porém, diante de ataques da falta de cooperação dos nós, o desempenho e a eficácia desses sistemas podem ser comprometidos;

b. *Sybil:* ocorre quando nós adversários criam múltiplas identidades falsas na rede enquanto utilizam um único dispositivo físico [25]. As identidades adicionais de um nó Sybil podem ser obtidas de duas formas: o nó Sybil pode fabricar uma nova identidade ou pode roubar uma identidade de um outro nó legítimo, o que é conhecido com personificação [46]. Esse tipo de ataque pode reduzir a eficácia de esquemas como armazenamento distribuído [21], roteamento multicaminhos [41], mecanismos baseados em confiança ou eleição [60], esquemas de reputação [15] e outros. Esses sistemas, que confiam na redundância de informações, devem garantir que as identidades estejam relacionadas com entidades distintas [2].

Um exemplo de ataque *Sybil* em que um dado nó x_m possui uma identidade adicional, chamada x_{m2}, é ilustrado na Figura 1.1. Nesse exemplo, o nó x_a precisa montar duas cadeias de confiança disjuntas até o nó x_i. Como o nó *Sybil* possui duas identidades, o nó x_a pode obter as cadeias $(x_a \rightarrow x_m \rightarrow x_e \rightarrow x_i)$ e $(x_a \rightarrow x_{m2} \rightarrow x_d \rightarrow x_f \rightarrow x_i)$. Assim, embora essas cadeias pareçam disjuntas, as informações referentes aos nós x_d e x_e foram fornecidas pelo nó *Sybil*, e podem comprometer a redundância do sistema.

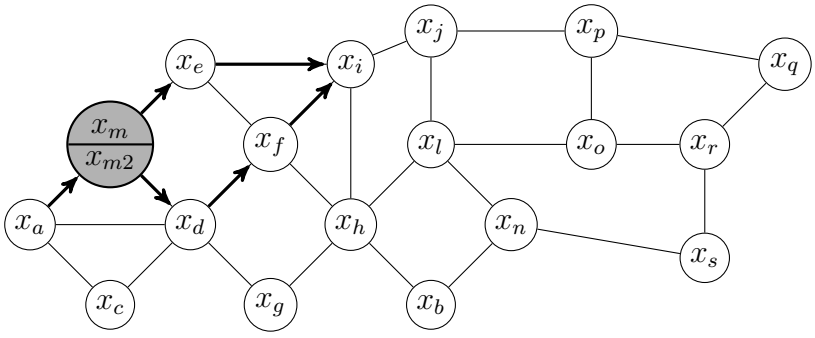


Figura 1.1: Ataque *Sybil* em um esquema de confiança baseado em cadeias

Um ataque *Sybil* também pode ser utilizado para comprometer os serviços de gerenciamento de chaves [64]. Diversos esquemas de gerenciamento de chaves para MANETs, por

exemplo, utilizam o conceito de certificação em grupo [38, 37] ou de cadeia de certificados [9, 11], visando amenizar os requisitos de uma Infraestrutura de Chaves Públicas (ICP) centralizada. Nesses casos, se um atacante *Sybil* possuir uma grande quantidade de identidades distintas, ele pode controlar efetivamente o esquema de gerenciamento de chaves da rede, permitindo a entrada de outros nós no sistema [51].

1.3 Objetivos e contribuições

Este trabalho propõe um esquema de gerenciamento de chaves públicas para MANETs baseado em grupos, que atuará como um *middleware* fornecendo serviços de gerência de chaves para as aplicações de um sistema. Esse esquema, chamado de *Survivable Group-based Public Key Management for MANETs* (SG-PKM), tem como objetivo manter o seu desempenho e eficácia mesmo diante de ataques de falta de cooperação e *Sybil*, minimizando o impacto da criação de identidades falsas na rede. No SG-PKM os nós formam grupos baseado nas relações de amizades de seus respectivos usuários. Além disso, como os nós podem participar de vários grupos ou os usuários podem ter relações de amizades com membros de outros grupos, é possível que os membros de um grupo forneçam informações sobre outros grupos, possibilitando que membros de grupos distintos possam autenticar-se mutuamente.

O SG-PKM é totalmente distribuído e auto-organizado, não sendo necessária uma entidade central para o gerenciamento das informações nem mesmo antes da formação da rede, e todos os nós podem fornecer os serviços de gerência de chaves. Além disso, as operações de formação dos grupos, emissão de certificados, autenticação das chaves públicas, revogação e atualização dos certificados baseiam-se na redundância de informações, sendo necessário mais que um nó para a confirmação de uma operação. Com essas características básicas, o SG-PKM se apresenta mais resistente aos ataques *Sybil* do que o principal esquema de gerência de chaves totalmente distribuído e auto-organizado para MANETs: o PGP-*Like*.

Este trabalho apresenta as seguintes contribuições:

a) levantamento das fraquezas e vulnerabilidades do PGP-*Like*;

b) criação de métricas para a quantificação do impacto dos ataques de má-conduta no PGP-*Like*;

c) avaliação do PGP-*Like* em cenários com ataques de falta de cooperação e *Sybil*;

d) proposta de um novo esquema de gerenciamento de chaves que mantenha o seu desempenho diante de ataques de falta de cooperação e seja mais resistente que o PGP-*Like* aos ataques *Sybil*;

e) análise das redes sociais de amigos para avaliar a possibilidade das formações de grupos e dos relacionamentos entre os grupos;

f) definição de métricas para a avaliação do SG-PKM diante dos ataques de falta de cooperação e *Sybil*;

g) avaliação do SG-PKM em cenários com ataques de falta de cooperação e *Sybil*.

1.4 Organização do texto

Este trabalho está organizadas em cinco capítulos. O Capítulo 2 discute as características desejáveis de um esquema de gerenciamento de chaves para as MANETs. Em seguida, aborda o *Sistema de Gerenciamento de Chaves Públicas Auto-Organizado para MANETs*, chamado neste trabalho de PGP-*Like*. Inicialmente, as características e as vulnerabilidades deste sistema são discutidas. Em seguida, é apresentada a quantificação dos ataques de falta de cooperação e de personificação no PGP-*Like*.

O Capítulo 3 descreve a proposta de um esquema de gerenciamento de chaves totalmente distribuído e auto-organizado que reduz o impacto dos ataques de falta de cooperação e *Sybil*. É apresentada uma visão geral do funcionamento do esquema de gerenciamento de chaves proposto e as suas principais operações como: a entrada dos nós na rede e a criação dos grupos; as trocas dos certificados de grupos; a atualização e a revogação dos certificados; e a autenticação das chaves públicas.

O Capítulo 4 apresenta uma avaliação do SG-PKM. Inicialmente é realizada uma análise dos relacionamentos entre os amigos em um repositório de certificados *Pretty Good Privacy* (PGP), com o objetivo de avaliar as possibilidades de formação dos grupos e dos relacionamentos entre esses grupos em uma rede real. Em seguida, uma avaliação dos custos de comunicação adicionados pelo SG-PKM nas operações de atualização, revogação e autenticação dos certificados é apresentada. Por fim, são apresentados os resultados da quantificação do impacto dos ataques de falta de cooperação e *Sybil* no SG-PKM, usando métricas definidas neste capítulo.

Por fim, o Capítulo 5 contém as conclusões do trabalho, com uma discussão das contribuições e resultados obtidos, e a relação dos trabalhos futuros.

CAPÍTULO 2

GERENCIAMENTO DE CHAVES EM MANETS

O serviço de gerenciamento de chaves é um conjunto de técnicas e procedimentos para suportar o estabelecimento e a manutenção das chaves utilizadas entre as partes autorizadas de uma comunicação [43]. Esse serviço é responsável por manter as relações criptográficas e o material criptográfico, que são os pares de chaves pública e privada, os parâmetros de inicialização e os parâmetros não secretos para o suporte ao gerenciamento de chaves. Além disso, deve considerar ameaças como o comprometimento da confidencialidade e da autenticidade das chaves públicas e privadas e o uso não autorizado dessas chaves [59]. Os sistemas de gerenciamento de chaves devem fornecer os seguintes serviços [43]:

a. inicialização dos usuários do sistema dentro da rede;

b. geração, distribuição e instalação do material criptográfico;

c. controle do uso do material criptográfico;

d. armazenamento e recuperação do material criptográfico;

e. inicialização e manutenção da confiança no material criptográfico.

Nas MANETs, o gerenciamento de chaves deve considerar, também, a mobilidade dos nós e a topologia dinâmica da rede, e deve, preferencialmente, ser auto-organizado e descentralizado [34, 62]. Além disso, um modelo robusto de gerenciamento de chaves para MANETs precisa satisfazer requisitos básicos como: não ter um ponto único de falha; ser tolerante a comprometimentos; ser capaz de revogar as chaves dos nós comprometidos e atualizar as chaves dos nós não-comprometidos; ser eficiente quanto ao armazenamento, o processamento e a comunicação [43].

É possível classificar os esquemas de gerenciamento de chaves para as MANETs em centralizados, parcialmente distribuídos ou totalmente distribuídos. Nos esquemas de

gerenciamento de chaves **centralizados**, assume-se a existência de uma entidade central, chamada Terceira Entidade Confiável (*Third Trusted Party* (TTP)), para a realização de todas ou algumas das funções. Essa TTP pode ser um nó pertencente à rede (TTP *online*) ou um elemento externo (TTP *offline*). Geralmente, uma TTP *offline* é usada nos casos em que os nós são pré-configurados com algum material criptográfico antes de entrarem na rede. Alguns exemplos de esquemas centralizados são apresentados em [29, 13, 27, 12, 26].

Já nos esquemas **parcialmente distribuídos**, as funções do gerenciamento de chaves são distribuídas entre um conjunto de nós pertencentes à rede. Esse tipo de organização tem como objetivo minimizar o impacto do ponto único de falha (*Single Point of Failure* (SPOF)) presente nos esquemas centralizados. Os esquemas parcialmente distribuídos podem ser divididos em gerenciados ou auto-organizados.

Os esquemas **parcialmente distribuídos e gerenciados** consideram a falta da infraestrutura de servidores nas redes *ad hoc* e distribuem a funcionalidade de uma autoridade central entre um subgrupo de nós, que formam uma Autoridade Certificadora Distribuída (ACD). Com isso, esses esquemas oferecem maior segurança e disponibilidade, se comparados com as abordagens centralizadas. Porém, ainda assume-se a exitência de uma TTP *offline* para a distribuição do material criptográfico, antes da formação entre os nós que formam essa ACD. Alguns esquemas de gerenciamento de chaves parcialmente distribuídos e gerenciados propostos para as MANETs são encontrados em [73, 70].

Os esquemas **parcialmente distribuídos e auto-organizados**, por outro lado, não requerem a presença de qualquer TTP, *online* ou *offline*, nem mesmo na fase de formação da rede. Neste caso, o serviço de gerenciamento de chaves é realizado por um subconjunto de nós de uma forma totalmente auto-organizada. Um exemplo de esquema parcialmente distribuído e auto-organizado é encontrado em [71].

Por fim, nos esquemas **totalmente distribuídos**, as funções do gerenciamento de chaves são distribuídas entre todos os nós pertencentes à rede. Nesse caso, sempre que um novo nó entra na rede, ele deve receber material criptográfico e pode oferecer os serviços de gerenciamento de chaves para os demais nós. Esses esquemas também podem ser divididos em gerenciados e auto-organizados.

Os esquemas **totalmente distribuídos e gerenciados** necessitam de uma TTP, geralmente *offline*, para realizar a distribuição do material criptográfico ou a definição dos níveis de confiança entre os nós. Um exemplo de esquemas totalmente distribuídos e gerenciados é encontrado em [42].

Os esquemas **totalmente distribuídos e auto-organizados** não necessitam de qualquer tipo de entidade central nem mesmo na fase de formação da rede. O principal esquema de gerenciamento de chaves totalmente distribuído e auto-organizado é chamado de *Sistemas de Gerenciamento de Chaves Públicas Auto-organizado* e apresentado em [35, 9]. As características e vulnerabilidades desse esquema serão discutidas na próxima seção. Outros esquemas totalmente distribuídos e auto-organizados podem ser encontrados em [11, 10].

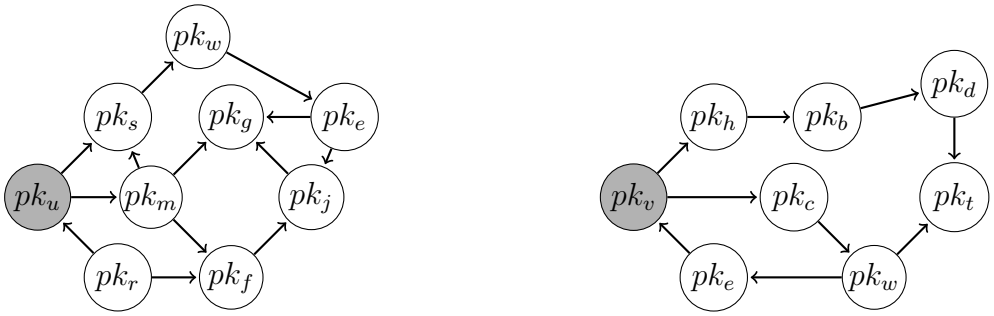
2.1 O sistema de gerenciamento de chaves públicas auto-organizado

O *Sistema de Gerenciamento de Chaves Públicas Auto-organizado* [9, 35], neste trabalho chamado de PGP-*Like*, é um esquema totalmente distribuído e auto-organizado que utiliza cadeias de certificados. Como ele é baseado nos conceitos do PGP [74], os próprios nós criam os seus pares de chaves públicas e privadas. Além disso, cada nó emite certificados de chaves públicas para os nós de sua confiança. Por outro lado, diferente do PGP, em que os certificados emitidos são armazenados em um repositório central, no PGP-*Like* esses certificados são armazenados e distribuídos pelos próprios nós de uma forma completamente auto-organizada e distribuída.

No PGP-*Like*, as chaves públicas e os certificados são denotados por um grafo direcionado $G = (V, A)$, no qual V representa as chaves públicas dos nós e A os certificados. Desta forma, uma aresta direcionada entre dois vértices pk_u e pk_v, representada por $(pk_u \rightarrow pk_v)$, denota um certificado assinado com a chave privada de x_u (sk_u), associando pk_v ao nó x_v. Além disso, um caminho conectando dois vértices, pk_u e pk_w ($pk_u \rightsquigarrow pk_w$), representa uma cadeia de certificados de pk_u até pk_w. Nessa cadeia, o primeiro certificado

pode ser verificado diretamente pelo nó x_u, e os demais certificados podem ser verificados usando a chave pública do certificado anterior na cadeia. Por fim, o último certificado contém a chave pública do nó x_w.

Para que um dado nó x_u possa autenticar corretamente um outro nó x_w via uma cadeia de certificados, ele deve garantir que todos os certificados na cadeia sejam válidos e corretos. Desta maneira, a validade de uma cadeia de certificados exige a honestidade de todos os nós presentes nela, o que não é fácil de se garantir nas MANETs [24]. Para construir cadeias de certificados apropriadas, cada nó x_u mantém dois repositórios locais, o repositório de certificados atualizados (G_u) e o de certificados não-atualizados (G_u^N). No repositório de certificados atualizados são armazenados os certificados que o nó x_u mantém sempre validados. Já no repositório de certificados não-atualizados são armazendos os certificados expirados e não atualizados pelos emissores ou aqueles certificados cuja validade ainda não tenha sido verificada. As Figuras 2.1a e 2.1b ilustram dois repositórios de certificados atualizados, pertencentes aos nós x_u e x_v respectivamente.



(a) Repositório $\mathbf{G_u}$ **(b)** Repositório $\mathbf{G_v}$

Figura 2.1: Repositórios de certificados atualizados dos nós x_u e x_v

Se um dado nó x_u precisa realizar uma comunicação segura com um outro nó x_v, e precisar certificar-se de que a chave pública pk_v pertecente ao nó x_v, ele precisa verificar a autenticidade dessa chave. Para isso, ele pode, por exemplo, obter um certificado válido associado a chave pública pk_v à identidade do nó x_v. Contudo, esse certificado precisa estar assinado por um nó confiável, para mitigar a ação de nós maliciosos. Depois que o nó x_u certifica-se de que a chave pública do nó x_v é válida, ambos os nós (x_u e x_v) podem criar uma chave uma chave de sessão para a realização da comunicação segura.

Quando um dado nó x_u deseja verificar a autenticidade da chave pública pk_v do nó x_v, ele busca, primeiramente, em seu repositório local de certificados atualizados (Figura 2.1a), um caminho do vértice pk_u até pk_v. Caso $\exists(pk_u \rightsquigarrow pk_v) \in G_u$, o nó x_u utiliza os certificados nesse caminho para autenticar pk_v. Caso $\nexists(pk_u \rightsquigarrow pk_v) \in G_u$, ele solicita ao nó x_v o seu repositório de certificados atualizados (G_v), e realiza a junção dos repositórios atualizados dos nós x_u e x_v, criando $G_1 = G_u \cup G_v$ (Figura 2.2). Nesse caso, a cadeia $(pk_u \rightarrow pk_s \rightarrow pk_w \rightarrow pk_e \rightarrow pk_v)$ pode ser utilizada pelo nó x_v para autenticar a chave pública pk_v do nó x_v.

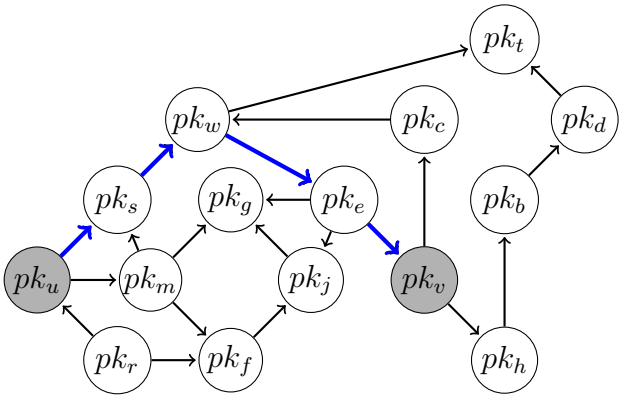


Figura 2.2: Repositório de certificados $G_u \cup G_v$ e os caminhos de certificados

Caso uma cadeia de certificados não tenha sido encontrada em G_1, o nó x_u cria $G_2 = G_u \cup G_u^N$ e procura $(pk_u \rightsquigarrow pk_v) \in G_2$. Se tal caminho for encontrado, o nó x_u precisa atualizar todos os certificados expirados ou não-atualizados, verificando suas autenticidades, junto aos seus respectivos emissores. Nesse caso, são necessárias trocas de mensagens adicionais entre o nó x_u e os emissores dos certificados. Por fim, se $\nexists(pk_u \rightsquigarrow pk_v) \in G_2$, ou a validade dos certificados contidos nessa cadeia não puder ser verificada, o nó x_u não realiza a autenticação de pk_v.

No PGP-*Like*, se um nó x_u acredita que uma dada chave pública pk_v pertence a um dado nó x_v, ele pode emitir um certificado assinado, com um tempo de validade $T_{validade}$, associando pk_v ao nó x_v, denotado por $(x_v, pk_v)_{pk_u}$. Cada nó também implementa um mecanismo de troca de certificados, que consiste em trocas periódicas de certificados com os seus vizinhos físicos. Em outras palavras, periodicamente cada vizinho físico do nó x_u

solicita a ele os seus repositórios de certificados, G_u e G_u^N. Assim, após sucessivas trocas de certificados e considerando a mobilidade dos nós, todos os certificados podem estar armazenados por todos os nós da rede. O período necessário para que um certificado alcance todos os nós é denominado tempo de convergência (T_{CE}). Logo, durante o T_{CE}, um nó pode encontrar problemas de autenticação, pois os certificados emitidos ainda não fazem parte dos repositórios locais de todos os nós da rede. Além disso, as trocas periódicas de certificados podem ser exploradas por ataques de falta de cooperação, nas quais os nós egoístas podem não cooperar com o PGP-*Like*, impedindo a propagação dos certificados emitidos.

Essas características, no entanto, tornam o sistema vulnerável a ataques de criação de falsas identidades, pois um nó malicioso não precisa comprometer outros nós para iniciar um ataque. Embora o PGP-*Like* permita que os nós detectem certificados inconsistentes e determinem qual associação "nó-chave" está correta, ele não impede que um atacante crie uma identidade falsa x_f, emita um certificado associando pk_f à x_f, e convença um usuário correto de que esse certificado é válido [71, 33, 32]. Nesse caso, se um nó malicioso mantiver um comportamento correto até o momento da criação das identidades falsas, quando ele iniciar o ataque, todas as identidades falsas poderão ser distribuídas pela rede, tornando-se parte de muitas cadeias de certificados.

Para evitar tal comportamento, o PGP-*Like* contém um mecanismo de detecção de certificados conflitantes. Se o nó x_u recebe um certificado contendo um associação "nó-chave" (x_v, pk_v) e ele não possui essa associação em qualquer outro certificado em G_u ou G_u^N, tal certificado é definido como *não-especificado*, visto que o nó u não possui informações suficientes para validá-lo. Se o nó x_u receber um outro certificado com a associação "nó-chave" (x_v, pk'_v) ou (f, pk_v), ele marca ambos os certificados como *conflitantes*. Caso o nó u não receba qualquer certificado conflitante em um certo período de tempo, ele classifica o certificado original como *não-conflitante*. Contudo, quando o nó x_u detecta um conflito, ele procura cadeias de certificados válidos e não-conflitantes para as chaves públicas pk_v e pk'_v. Baseado nessas cadeias, o nó x_u pode classificar um certificado como *não-conflitante* e o outro como *falso*. Se o nó x_u não conseguir tomar uma decisão, ambos certificados

continuam classificados como *conflitantes*. Segundo os autores do PGP-*Like*, esse método impede que um atacante emita um certificado associando pk_v ao nó falso x_f ou uma chave falsa pk'_v a um nó autêntico.

Por fim, o PGP-*Like* assume a existência de um modelo de confiança entre os nós e, com base nesse modelo, as cadeias de certificados são criadas. Essas cadeias representam a confiança existente entre os nós e são denominadas cadeias de confiança. Contudo, as cadeias de confiança apresentam uma autenticação fraca [17], pois elas assumem uma confiança transitiva, ou seja, se o nó x_a confia no nó x_b e o nó x_b confia no nó x_c, então o nó x_a também confia no nó x_c, e isso nem sempre pode ser considerado verdade.

2.2 Métricas para a avaliação do PGP-Like

Para avaliar a eficácia do PGP-*Like* diante dos ataques de falta de cooperação e *Sybil* foram usadas cinco métricas. Duas dessas métricas, Convergência das Trocas de Certificados (*Certificate Exchange Convergence* (CE)) e Alcançabilidade dos Usuários (*User Reachability* (UR)), são utilizadas por [9] na avaliação do PGP-*Like* em cenários sem ataques e serão usadas nesta avaliação para quantificar o impacto dos ataques de falta de cooperação. Já para avaliar o impacto dos ataques *Sybil*, são introduzidas três novas métricas, denominadas como Confiabilidade em uma Identidade Falsa (*False Identity Confidence* (FIC)), Autenticação Indireta de uma identidade falsa (*Indirect Authentication* (IA)) e Certificados Suspeitos por repositório (*Suspicious Certificates* (SC)). Tais métricas são importantes na avaliação do ataque *Sybil* no PGP-*Like* pois avaliam as cadeias de certificados nos repositórios de cada nó.

A notação utilizada na descrição das métricas é a mesma utilizada em [9]. Todas as métricas consideram S o conjunto de nós do sistema, $|X|$ o número de elementos de um conjunto X e NC o subconjunto de nós não-comprometidos.

- CE é a percentagem média dos certificados de G contidos nos repositórios locais dos nós no tempo t. Ela representa a convergência dos repositórios dos nós no tempo t e mostra o tempo necessário para que os certificados alcancem todos os nós do

sistema. CE é definida como:

$$CE(t) = \frac{\sum_{x_i \in S} CE_i(t)}{|S|} \qquad \text{em que}$$

$$CE_i(t) = \frac{\sum_{x_a, x_b \in S} (pk_a \rightarrow pk_b) \in (G_i \ \cup \ G_i^N)}{\sum_{x_x, x_y \in S} (pk_x \rightarrow pk_y) \in G} \tag{2.1}$$

- UR é a percentagem média de caminhos que os nós podem encontrar em seus repositórios locais de certificados, atualizados e não atualizados, no tempo t. Ela mostra a eficácia do mecanismo de troca de certificados para a autenticação das chaves. UR é definida como:

$$UR(t) = \frac{\sum_{x_i \in S} UR_i(t)}{|S|} \qquad \text{em que}$$

$$UR_i = \frac{\sum_{x_a \in S} (pk_i \rightsquigarrow pk_a) \in (G_i \ \cup \ G_i^N)}{|S|} \tag{2.2}$$

- FIC é o número de nós não-comprometidos que confiam em uma identidade falsa. Ela representa o tempo necessário para que uma identidade falsa se torne parte dos repositórios locais de todos os nós. FIC é definida como:

$$FIC = \frac{\sum_{x_i \in NC} FIC_i}{|NC|} \qquad \text{em que}$$

$$FIC_i = \begin{cases} 1 & \text{caso} \quad \exists \ x_f \in (G_i \cup G_i^N) \text{ sendo que } \ x_f \in (S \cap NC) \\ 0 & \text{caso contrário} \end{cases} \tag{2.3}$$

- IA é o índice de nós não-comprometidos (x_i) que autenticam uma identidade falsa (x_f), considerando a união dos repositórios atualizados locais do nó x_i (G_i) com o nó x_f (G_f). Ela representa a velocidade com a qual uma identidade falsa pode ser autenticada pelos nós não-comprometidos. IA é definida como:

$$IA = \frac{\sum_{x_i \in NC} IA_i}{|NC|} \qquad \text{em que}$$

$$IA_i = \begin{cases} 1 & \text{caso} \quad \exists (pk_i \rightsquigarrow pk_f) \in (G_i \cup G_f) \ \text{ sendo que } \ x_f \in (S \cap NC) \\ 0 & \text{caso contrário} \end{cases} \tag{2.4}$$

- SC é a fração de certificados emitidos por um nó *Sybil* encontrados nos repositórios locais dos nós não-comprometidos. Tais certificados podem estar ou não associados a uma identidade falsa. Entretanto, devido à ausência de um mecanismo de detecção de má-conduta, eles são considerados suspeitos. Sendo F o conjunto de nós *Sybil* no sistema, SC é definida como:

$$SC = \frac{\sum_{x_i \in NC} SC_i}{|NC|} \qquad \text{em que}$$

$$SC_i = \frac{\sum_{x_z \in G_i} \sum_{x_f \in F} (pk_z \rightarrow pk_f) \in G_i}{|G_i|} \tag{2.5}$$

2.3 Avaliação do PGP-Like diante de ataques de falta de cooperação e de *Sybil*

Na avaliação do impacto dos ataques de falta de cooperação e *Sybil* sobre o PGP-*Like*, foi utilizado o simulador NS versão 2.30 [48]. Como em [9], as simulação foram executadas usando grafos de certificados aleatórios, com trocas de certificados periódicas, a cada 60 segundos. Da mesma forma, assume-se que os nós não realizam as trocas de certificados simetricamente e que a rede não possui um mecanismo de detecção de má-conduta. Por simplicidade e sem detrimento dos resultados, nas simulações, os nós emitem os certificados e criam suas próprias chaves públicas e privadas apenas durante a formação da rede. São emitidos 600 certificados entre pares de nós selecionados aleatoriamente. O modelo de propagação utilizado é o reflexão no solo de dois raios (*two-ray ground reflection*) [52] e o protocolo de camada de enlace é o IEEE 802.11 [36]. Os demais parâmetros das simulações encontram-se na Tabela 2.1 e os resultados apresentados são médias de 35 simulações com intervalo de confiança de 95%.

2.3.1 Ataques de falta de cooperação

Na avaliação do PGP-*Like* em ambientes com ataques de falta de cooperação, foram considerados 5%, 10%, 20% e 40% de nós egoístas. Esses nós realizam todas as operações

Tabela 2.1: Parâmetros dos cenário das simulações

Parâmetro	**Valores utilizados**
Raio de alcance	50 e 120 metros
Quantidade de nós	100 nós
Tempo de vida da rede	10000 segundos
Velocidades máximas	5, 10 e 20 m/s
Tamanho do ambiente	1000 x 1000 e 1500 x 300 metros
Tipo de movimentação	*waypoint* aleatório
Tempo máximo de pausa	20 segundos
Tempo entre troca de certificados	60 segundos
Quantidade de certificados emitidos	600 certificados

básicas da rede, inclusive a emissão de certificados. Na fase de troca de certificados, eles solicitam e aceitam os certificados dos seus vizinhos, porém quando os vizinhos solicitam os seus certificados, eles não respondem. Note que, é possível autenticar os nós egoístas, pois qualquer nó pode emitir um certificado válido para ele.

Nesta seção, são apresentados os resultados das simulações realizadas em cenários de 1000 x 1000 metros. Devido à similaridade de comportamento, os resultados das simulações realizadas com cenários 1500 x 300 metros encontram-se no Apêndice A. A mudança no tamanho do ambiente afeta apenas o tempo necessário para a convergência das trocas de certificados, mas apresenta o mesmo padrão de comportamento.

A Figura 2.3 mostra a convergência das trocas de certificados ($CE(t)$) em cenários com o raio de alcance das antenas igual a 120 metros. Como esperado, o aumento da quantidade de atacantes resulta em uma queda em CE. De fato, aumentando a quantidade de nós egoístas, menos nós participam das trocas de certificados, o que afeta a quantidade de certificados nos repositórios locais e o tempo de convergência. Pode-se observar que o impacto é mínimo na presença de 5% a 20% de nós egoístas. Já com a percentagem de atacantes maior que 40%, esse impacto aumenta, e CE alcança um valor máximo de aproximadamente 83%.

Também é possível notar que, com o aumento da velocidade máxima de movimentação dos nós, é necessário um tempo menor para a convergência das trocas de certificados. Em cenários com 40% de nós egoístas e com velocidade máxima de 20 m/s, o valor CE estabiliza após 1100 segundos do tempo de vida da rede. Já com a velocidade máxima

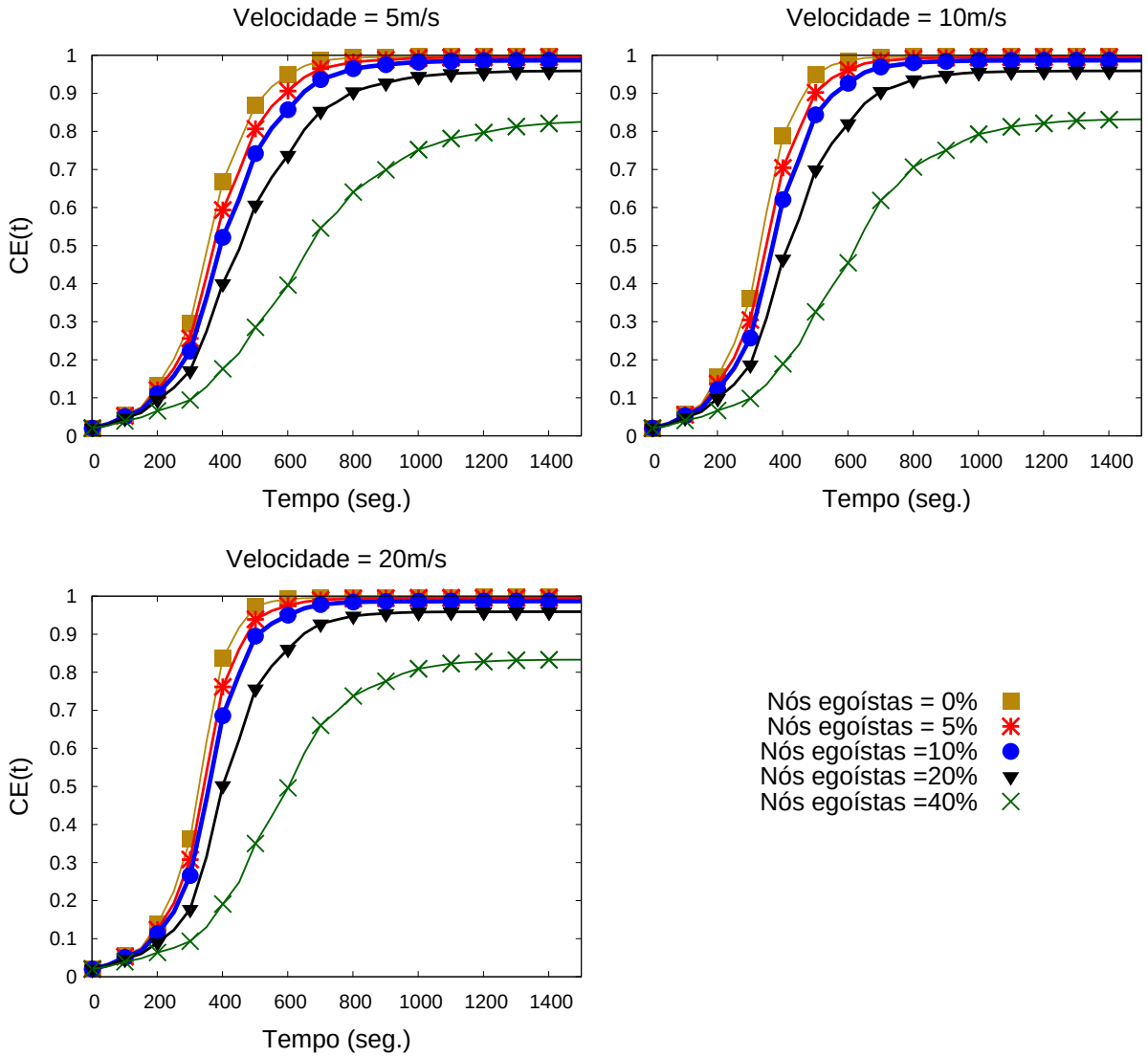


Figura 2.3: Convergência das trocas de certificados diante de ataques de falta de cooperação (1000 x 1000 metros e raio de 120 metros)

de 5 m/s, é preciso aproximadamente 1400 segundos para que esse valor estabilize, i.e um tempo 27% maior.

A Figura 2.4 mostra a percentagem média de certificados nos repositórios locais dos nós e o momento de convergência das trocas de certificados em cenários com o raio de alcance das antenas igual a 50 metros.

A diminuição do raio de alcance das antenas dos nós afeta o tempo necessário para a troca dos certificados na rede. Com um raio de alcance menor, os nós possuem menos vizinhos físicos e, dessa forma, trocam certificados com uma quantidade menor de nós. Assim, é necessário um tempo maior, e uma quantidade maior de trocas de mensagens, para que um certificado possa fazer parte de todos os repositórios locais dos nós. Mesmo

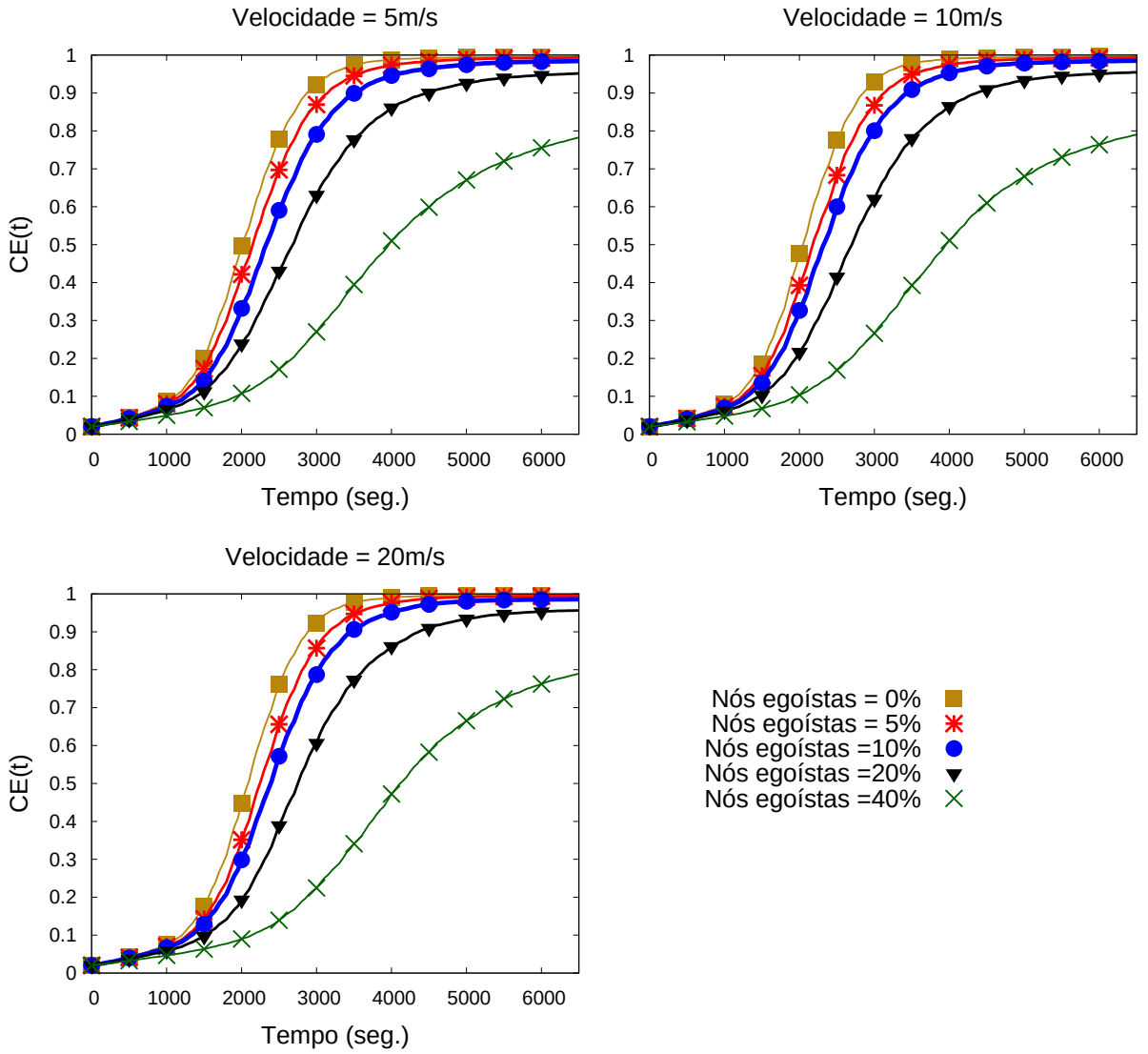


Figura 2.4: Convergência das trocas de certificados diante de ataques de falta de cooperação (1000 x 1000 metros e raio de 50 metros)

em cenários sem ataques, o valor de CE alcança 100% após aproxidamente 3500 segundos do tempo de vida da rede. Também, a velocidade de movimentação dos nós gera um leve impacto no tempo necessário para que CE alcance o seu ponto máximo.

A Figura 2.5 mostra a alcançabilidade dos nós (UR) usando os repositórios locais de certificados, em cenários de tamanho 1000 x 1000 metros e raio de alcance das antenas igual a 120 metros. Contradizendo os resultados teóricos encontrados em [33, 68], $UR(t)$ foi pouco afetada, mesmo diante de 40% de nós egoístas.

Após o tempo de convergência, mesmo na presença de até 40% de atacantes, a alcançabilidade dos nós é a mesma que a encontrada em cenários sem ataques. Em todos esses casos $UR(t)$ é aproximadamente 100%, demonstrando que, embora $CE(t)$ seja com-

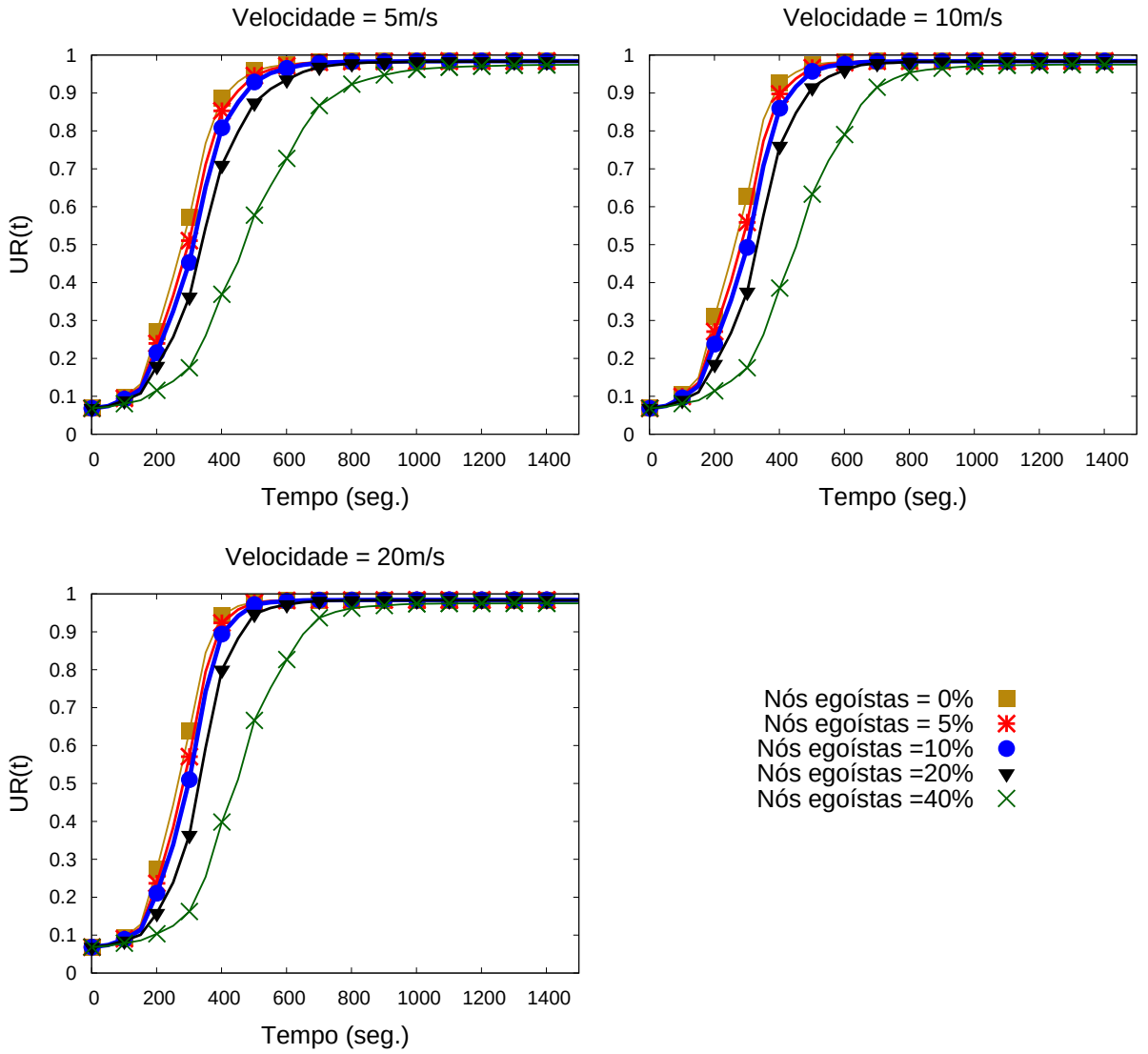


Figura 2.5: Alcançabilidade dos nós diante de ataques de falta de cooperação (1000 x 1000 metros e raio de 120 metros)

prometida diante de nós egoístas, a eficácia do PGP-*Like* pode ser garantida mesmo na presença de nós maliciosos.

Por fim, a Figura 2.6 também apresenta a alcançabilidade dos nós usando os repositórios locais de certificados, porém com o raio de alcance das antenas igual a 50 metros. Como no caso anterior, após o tempo de convergência, que nesse caso é maior devido à diminuição do raio de alcance das antenas, o valor de UR sempre atinge 100%.

Resumindo, independente do tamanho da rede ou do raio de alcance dos nós, mesmo na presença de até 40% de nós egoístas, o alcançabilidade dos nós (UR) não é afetada. Contudo, é importante salientar que $CE(t)$ tem um impacto direto na conectividade dos repositórios locais (G_u e G_u^N) de todos os nós. Quanto menor o valor de $CE(t)$, menos

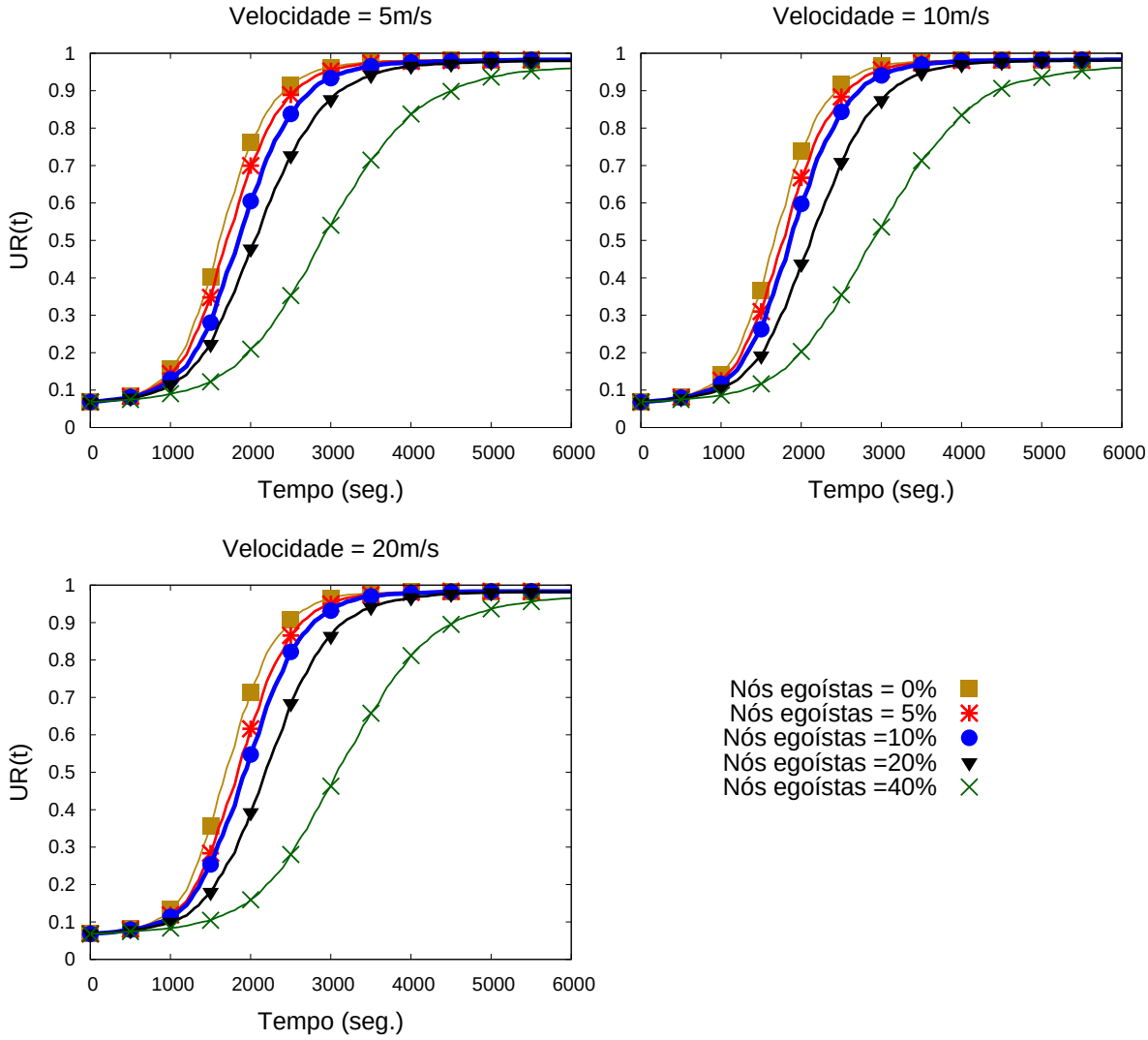


Figura 2.6: Alcançabilidade dos nós diante de ataques de falta de cooperação (1000 x 1000 metros e raio de 50 metros)

caminhos são encontrados em G_u e G_u^N. Assim, ter quase 100% em $UR(t)$, e um valor de $CE(t)$ de aproximadamente 83% (com aproximadamente 40% de nós egoístas), significa que quase todos os nós podem construir cadeias de certificados válidas para os demais nós. Mas a alcançabilidade dos nós estará limitada à validade das cadeias de certificados não-comprometidas ou à permanência na rede dos nós que as contenham.

2.3.2 Ataques Sybil

Na avaliação do PGP-*Like* em ambientes com ataques *Sybil*, foram considerados 5%, 10%, 20% e 40% de nós maliciosos. Dois comportamentos diferentes de nós *Sybil* foram analisados. No primeiro caso, os nós *Sybil* possuem um mau comportamento desde a

formação da rede, emitindo certificados que podem ser falsos ou não. No segundo caso, os nós *Sybil* têm um comportamento correto durante a fase inicial da rede, isto é, até a convergência do valor de CE. Após esse período, cada nó *Sybil* cria identidades falsas e emite certificados para elas, agindo maliciosamente no restante do tempo. Todos os parâmetros de simulação são os mesmos usados na avaliação em cenários com ataques de falta de cooperação (Seção 2.3.1).

As Figuras 2.7 e 2.8 apresentam o tempo necessário para que as identidades falsas sejam disseminadas pelo sistema, i.e. o tempo necessário para que todos os nós não-comprometidos possuam ao menos uma identidade falsa em seu repositório local de certificados. Essas figuras contém os resultados com velocidades máximas de 5 m/s, 10 m/s e 20 m/s, e com os seguintes cenários: ambiente de 1000m x 1000m e raio de alcance de 120m; ambiente de 1000m x 1000m e raio de alcance de 50m; ambiente de 1500m x 300m e raio de alcance de 120 metros; ambiente de 1500m x 300m e raio de alcance de 50m.

Na Figura 2.7 é considerado o primeiro caso, em que os nós *Sybil* emitem certificados falsos desde a formação da rede. Em todos os cenários apresentados, quanto maior a quantidade de nós maliciosos na rede, menos tempo é necessário para que as identidades falsas sejam propagadas pela rede. Com um raio de alcance de 120 metros e ambiente de 1000 x 1000 metros, em cenários com 5% de nós maliciosos é necessário aproxidamente 450 segundos para que todos os nós não-comprometidos tenham ao menos uma identidade falsa em seus repositórios locais. Já com 40% de nós maliciosos, esse tempo é de aproxidamente 370 segundos, i.e aproxidamente 17% a menos.

Em cenários com o mesmo tamanho de ambiente mas com raio de alcance igual a 50 metros, o tempo necessário para a disseminação das identidades falsas é sempre maior. Esse comportamento é esperado, visto que, nesses cenários, também é necessário um tempo maior para a convergência. No entanto, como no cenário anterior, na medida em que aumenta a quantida de nós maliciosos, diminui o tempo necessário para que os nós não-comprometidos sejam infectados.

Na Figura 2.8 é considerado o segundo caso, em que os nós *Sybil* emitem certificados falsos após a convergência das trocas de certificados. Nesse caso, é apresentado o tempo

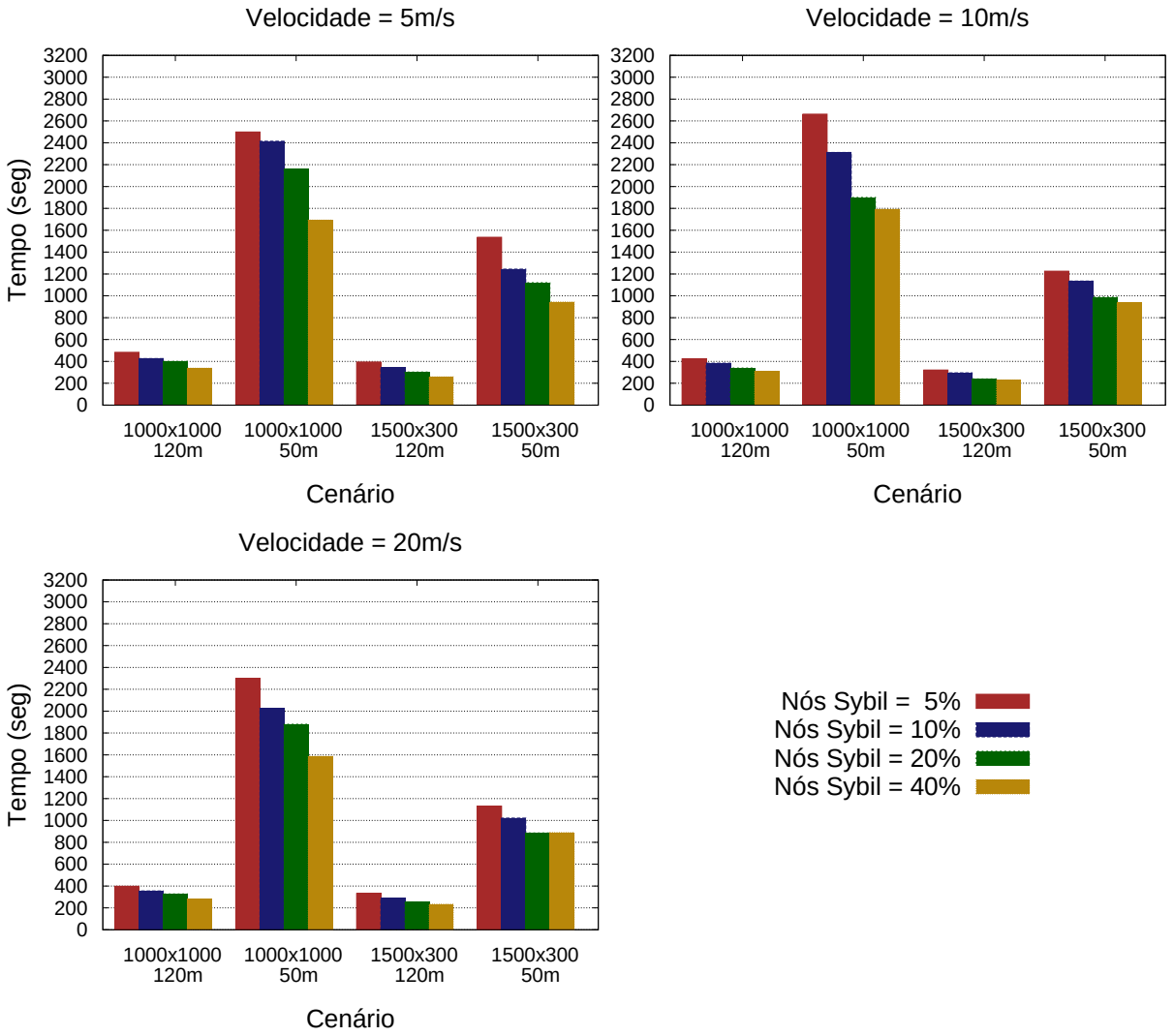


Figura 2.7: Confiabilidade em identidades falsas criadas na formação da rede

necessário, após o início dos ataques, para que todos os nós não-comprometidos do sistema tenham ao menos uma identidade falsa em seu repositório local de certificados. Da mesma forma que no primeiro caso, quanto maior a quantidade de nós maliciosos na rede, menos tempo é necessário para que os nós sejam infectados. Com um raio de alcance de 120 metros e ambiente de 1000 x 1000 metros, em cenários com 5% de nós maliciosos é necessário aproxidamente 600 segundos para que o valor de FIC alcance 100%. Já com 40% de nós maliciosos, esse tempo é de aproxidamente 360 segundos, i.e cerca de 36% a menos.

Nos cenários com o tamanho de ambiente igual a 1500 x 300 metros o comportamento é similar, mas é necessário um tempo menor que FIC alcance a 100%. Nesse caso, com

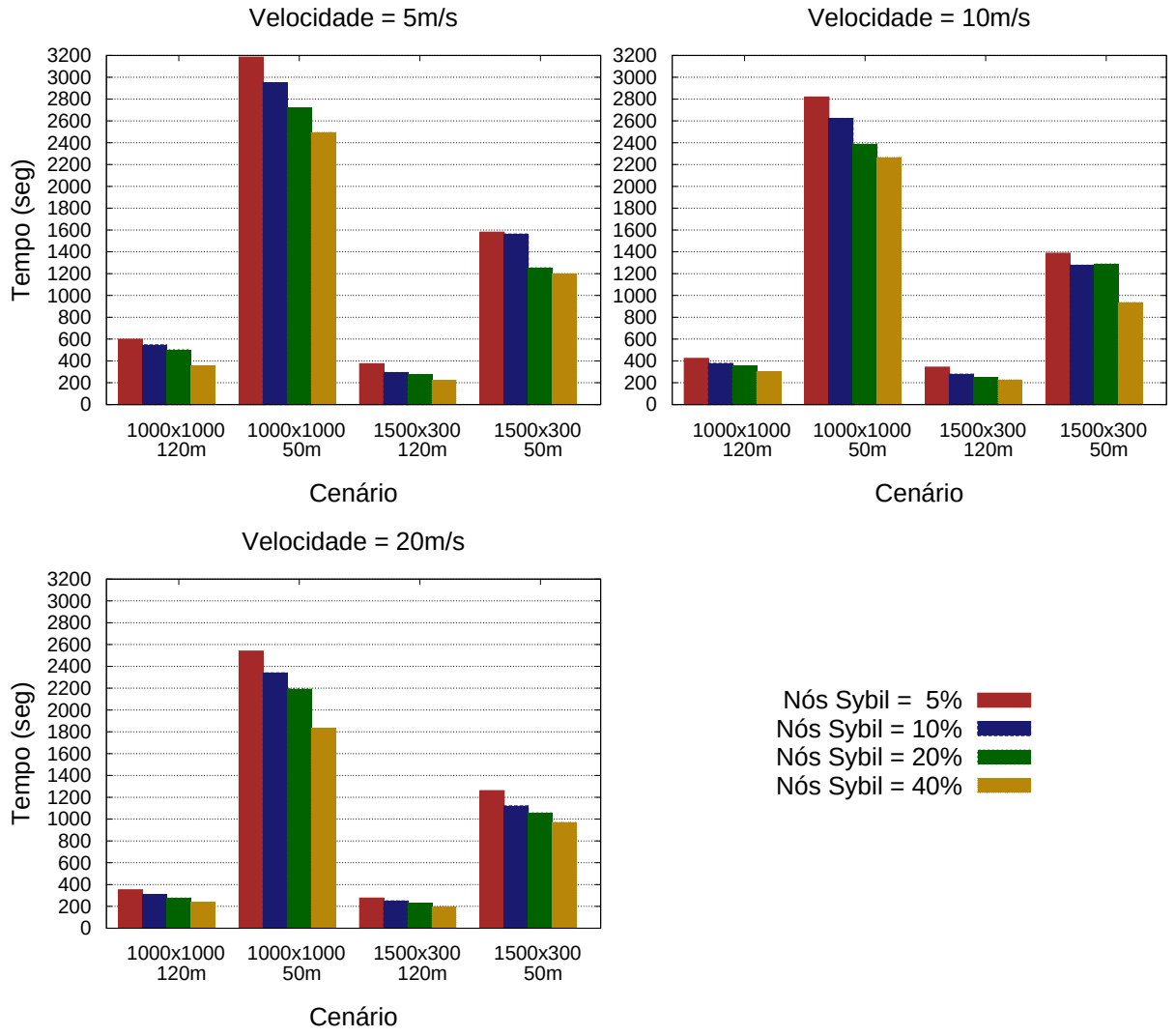


Figura 2.8: Confiabilidade em identidades falsas criadas após a convergência da rede

velocidades de 20 m/s, raio de alcance igual a 50 metros e 40% de nós *Sybil* é necessário, aproximadamente, 1200 segundos para que todos os nós não-comprometidos tenham ao menos uma identidade falsa em seus repositórios locais. Para os mesmos cenários, porém com raio de alcance de 120 metros, esse tempo é de apenas 200 segundos.

Resumidamente, tanto nos ataques que iniciam na formação da rede, como nos que iniciam após a convergência das trocas de certificados, os nós não-comprometidos são rapidamente infectados por no mínimo uma identidade falsa. Essa característica, que em situações normais é essencial para garantir a eficiência do sistema de gerenciamento, na presença de ataques *Sybil*, torna o sistema altamente vulnerável. Também, quando o raio de alcance é menor, é necessário um tempo maior para que FIC alcance 100%. Isso ocorre pois é necessário um tempo maior para a convergência das trocas de certificados.

Para que um nó *Sybil* utilize a sua identidade falsa x_f, ele deve apresentar um certificado emitido para essa identidade falsa, e tentar persuadir um nó não-comprometido a autenticar esse certificado. Nesse caso, o nó não-comprometido x_u procura por uma cadeia de certificados que valide a identidade falsa em G_u ou $G_u \cup G_f$. Se um ataque inicia após a convergência da rede, a taxa de autenticação mútua dos nós é aproximadamente 100%. Nesse caso, os nós *Sybil* conseguem convencer praticamente todos nós não-comprometidos da autenticidade da identidade falsa.

A Figura 2.9 apresenta o percentual de autenticação indireta (IA), que corresponde às cadeias de certificados contidas em $G_u \cup G_f$ que validam uma identidade falsa. Nas simulações, os nós *Sybil* criam identidades falsas logo após a convergência da rede, e o valor de IA é calculado no exato momento da criação dessas identidades. Independente do número de nós maliciosos, o valor de IA é sempre 100%, ou seja, todos os nós não-comprometidos são afetados e podem ser levados a autenticar uma identidade falsa.

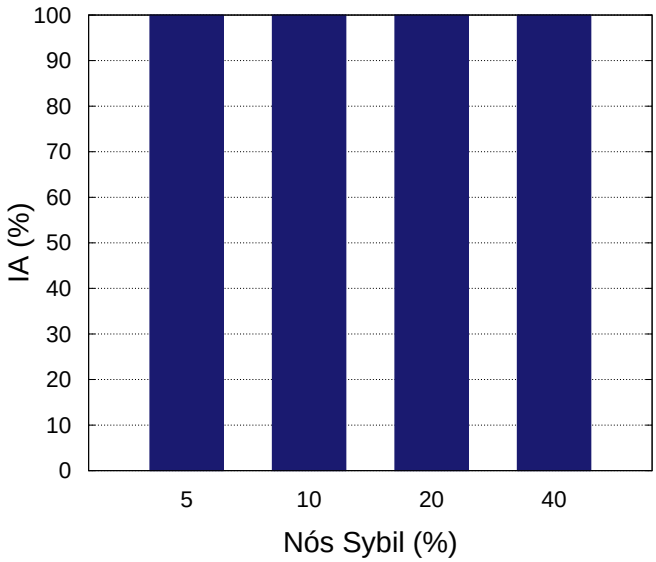


Figura 2.9: Percentual de identidades falsas autenticadas via Autenticação Indireta

Considerando os repositórios locais de um nó x_u, SC_u é a quantidade de cadeias de certificados que possuem no mínimo um certificado emitido por um nó *Sybil*. A Figura 2.10 mostra o número de cadeias de certificados afetadas após a convergência das trocas de certificados. É possível notar que o valor de SC aumenta com o número de nós maliciosos. Em cenários com 5% de nós maliciosos, SC é quase 45%, enquanto em cenários com 40%

de nós maliciosos, este valor chega a 70%.

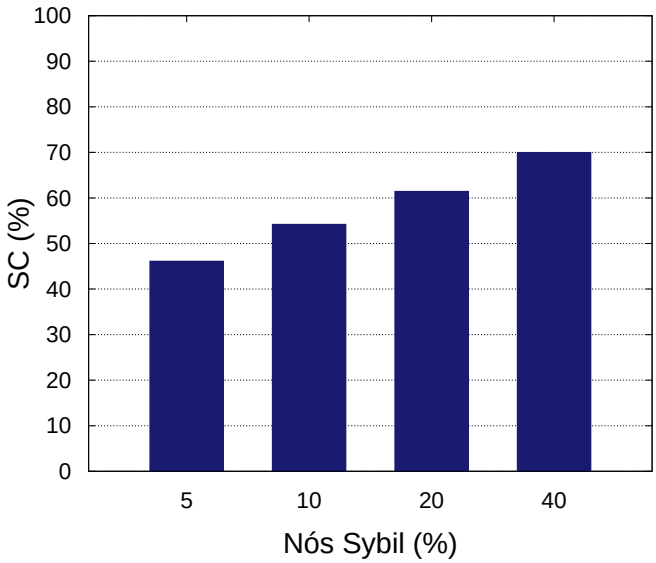


Figura 2.10: Percentual de Certificados Suspeitos nos repositórios locais de certificados

2.4 Conclusão

Neste capítulo foram apresentadas as características, funcionalidades e vulnerabilidades do PGP-*Like*, um esquema de gerenciamento de chaves totalmente auto-organizado e distribuído para MANETs. Como discutido, no PGP-*Like* cada nó cria o seu par de chaves pública e privada, e emite certificados para os nós de sua confiança. Os certificados emitidos são trocados entre os nós vizinhos e, eventualmente, farão parte dos repositórios de certificados locais de todos os nós.

Além disso, no PGP-*Like*, as autenticações mútuas são realizadas por meio de cadeias de certificados entre os nós. Esses dois nós não precisam ser vizinhos físicos, precisando, nesse caso, dos serviços de roteamento da fornecidos pela rede. Por outro lado, essas cadeias de certificados precisam da cooperação dos nós para se formarem, e podem apresentar uma autenticação fraca, facilitando a ação de um ataque *Sybil*, pois um atacante precisa comprometer apenas um elemento da cadeia para poder participar do sistema. Para avaliar o impacto dos ataques de falta de cooperação e *Sybil* no PGP-*Like* foram definidas métricas para a quantificação do impacto dos ataques de falta de cooperação e *Sybil* no PGP-*Like*.

Os resultados das simulações mostram que o PGP-*Like* é resistente aos ataques de falta de cooperação, mas totalmente vulnerável aos ataques *Sybil*, mesmo na presença de poucos nós maliciosos (5%). Assim, essa avaliação confirma os resultados teóricos [71, 33, 32], demonstrando que os ataques *Sybil* são grandes ameaças ao PGP-*Like* e que a sua eficácia fica comprometida independentemente do número de atacantes, sendo assim, necessário um mecanismo de segurança que reduza o impacto de tais ataques.

CAPÍTULO 3

ESQUEMA DE GERENCIAMENTO DE CHAVES PÚBLICAS SOBREVIVENTE BASEADO EM GRUPOS

Este capítulo apresenta um novo esquema de gerenciamento de chaves totalmente distribuído e auto-organizado, chamado de SG-PKM, que é mais resistente que o PGP-*Like* aos ataques de falta de cooperação e *Sybil*. Nesse novo esquema, os nós são organizados em grupos visando dificultar a ação dos nós maliciosos. Ele fornece serviços para as aplicações de um sistema, de modo que qualquer serviço que necessite de um serviço seguro de gerência de chaves pode utilizá-lo.

As próximas seções apresentam as características e o funcionamento do SG-PKM. A Seção 3.1 apresenta uma visão geral do funcionamento do esquema. As próximas seções discutem as operações do SG-PKM: a entrada de nós e a formação dos grupos (Seção 3.2); a criação e distribuição das chaves dos certificados de grupos (Seção 3.3); a emissão e distribuição dos certificados (Seções 3.5 e 3.4) a troca dos certificados e grupos e a construção dos repositório atualizados (Seção 3.6); os processos de autenticação (Seção 3.7), validação (Seção 3.8), atualização (Seção 3.9) e revogação (Seção 3.10) dos certificados de nós e grupos. Por fim, a Seção 3.11 apresenta a arquitetura utilizada como suporte à implementação do SG-PKM.

3.1 Visão geral

Alguns objetivos foram definidos quanto às características do novo esquema de gerenciamento de chaves públicas para MANETs, sendo eles:

a) ser totalmente distribuído;

b) ser auto-organizado;

c) manter o desempenho na ataques de falta de cooperação;

d) ser eficaz diante de ataques Sybil.

No SG-PKM, os nós formam grupos baseados na relação de confiança mútua entre eles. Essa confiança entre os nós depende das relações de amizade existentes entre os usuários participantes da rede. Se dois usuários, i e j, são amigos, então eles confiam um no outro e podem permitir que os seus respectivos dispositivos (nós), x_i e x_j, troquem as suas chaves públicas por um canal seguro, como um canal de comunicação infravermelho ou um *smart card*, em algum encontro físico. Neste trabalho assume-se uma confiabilidade bidirecional na qual, se i confia em j, então j também confia em i. Essa suposição é baseada em estudos discutidos em [72]. Nesses estudos, Zhang, Song e Fang fazem uma análise estatística do "*Web of Trust*" entre os usuários do PGP [1], e mostram que cerca de 2/3 das ligações em uma grande rede social fortemente conectada são bidirecionais.

As relações de amizades formam redes espontâneas [30], que são independentes da rede física e apresentam propriedades das redes sociais [19], como os fenômenos *small-world* e *scale-free*. O fenômeno *small world* é encontrado nas redes sociais em que cada par de usuários pode ser alcançado por meio de uma pequena cadeia de conhecimentos sociais [66, 69]. Já o fenômeno *scale-free* resulta da existência de poucos usuários com um número muito maior de amigos do que os outros [5, 4]. Além disso, esses poucos usuários têm uma grande probabilidade de serem escolhidos pelos novos usuários como seus amigos: paradigma conhecido como "*rich gets richer*" [65].

Com base nesse relacionamento de amizade, os nós formam grupos, em que todos os membros de um grupo servirão como testemunhas na criação de certificados de chaves públicas para os demais membros do grupo. Além disso, também baseado no mesmo relacionamento de amizade, esses grupos poderão se inter-relacionar, emitindo certificados mutuamente. O inter-relacionamento entre os grupos forma um grafo de certificados de grupos.

O SG-PKM pode ser visualizado como um modelo em três camadas, como ilustrado na Figura 3.1. As três camadas representam, respectivamente, os modelos de rede, de confiança e de grupos, e cada um desses modelos pode ser representado por um grafo. O **modelo de rede** representa as ligações físicas entre dois nós e pode ser denotado

por um grafo $G_{fis} = (V, A_{fis})$, no qual V representa o conjunto de nós do sistema e A_{fis} as ligações físicas entre esses vértices. Nesse caso, dois vértices vértices x_i e x_j são considerados vizinhos $(x_i, x_j) \in A_{fis}$, se a distância Euclidiana entre esses dois nós for menor do que o raio de alcance de suas respectivas antenas. Neste trabalho assume-se que as antenas de todos os nós possuem o mesmo raio de alcance.

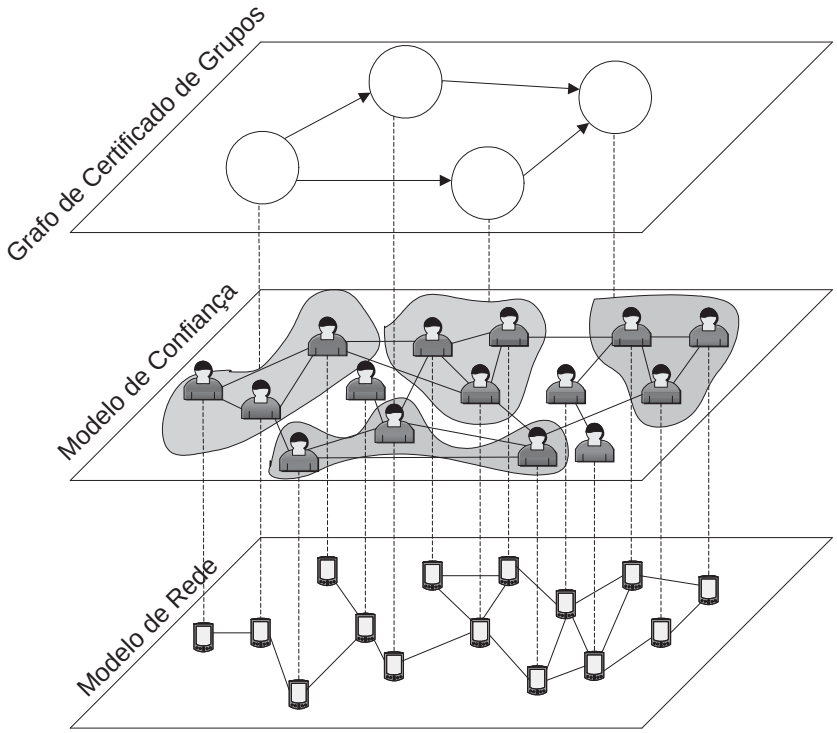


Figura 3.1: Visualização em camadas do SG-PKM

O **modelo de confiança**, representa as relações de confiança e amizade existente entre os nós. Esse modelo pode ser denotado pelo grafo direcionado $G_{con} = (V, A_{con})$, no qual V representa os nós do sistema e A_{con} representa as relações de confiança entre os nós. Nesse caso, um aresta $(x_i, x_j) \in A_{con}$ significa que o nó x_i confia no nó x_j, e vice-versa.

Por fim, o **modelo de grupo** representa os grupos formados pelos nós do sistema. O modelo de grupo também é denotado por um grafo direcionado $G = (IG, E)$, no qual IG representa o conjunto dos grupos formados no sistema, e E representa os certificados que foram emitidos entre os grupos. A formação desses grupos e o relacionamento entre eles são as bases para o funcionamento e eficácia do SG-PKM e serão melhor detalhados nas

próximas seções.

Assumindo esse modelo em três camadas, no SG-PKM cada nó cria seu próprio par de chaves pública e privada e, para participar do sistema de gerenciamento de chaves, ele precisa formar um grupo (IG) com outros ($m-1$) nós. Nesses pequenos grupos de tamanho m, todos os nós possuem o mesmo papel e não é necessária a presença de um líder, diferente de outras abordagens que usam o conceito de grupos [22, 47]. A Figura 3.2 ilustra dois grupos, IG_1 e IG_2. O grupo IG_1 é formado pelos nós x_1, x_2, x_3, x_4, x_5, x_6, x_7 e x_8, e o grupo G_2 é formado pelos nós x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13} e x_{14}. Nesse exemplo, os nós x_7 e x_8 fazem parte dos dois grupos, IG_1 e IG_2, e podem servir como testemunhas da autenticidade dos dois grupos.

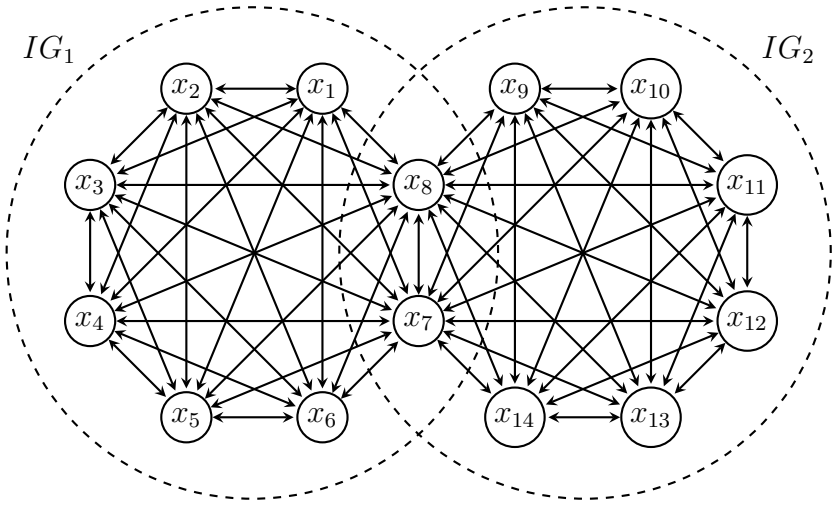


Figura 3.2: Representação dos grupos

Os nós que iniciam a rede trocam as suas chaves públicas por um canal seguro, baseados em seus relacionamentos de amizade, e formam os primeiros grupos do sistema. A partir daí, os novos nós que entram no sistema precisam trocar informações com os demais nós, para poderem, então, criar novos grupos e participarem das atividades da rede.

Na formação de um dado grupo IG_α, os nós que participam desse grupo constroem, colaborativamente, o par de chaves pública e privada do grupo, representadas respectivamente por PK_α e SK_α. Esse par de chaves pode ser construído utilizando qualquer esquema de acordo de chaves distribuído e sem uma TTP, por exemplo [50, 14, 31]. Após a criação das chaves, a chave pública do grupo é disponibilizada a todos os nós da rede

e a chave privada é distribuída entre os m nós participantes do grupo seguindo um esquema de criptografia de limiar (t, m) [56]. Também, são emitidos certificados associando a chave pública de cada nó pertencente ao grupo com a sua respectiva identidade. Esses certificados, chamados de *certificados dos nós*, são assinados com a chave privada do grupo.

Os grupos também podem emitir certificados entre si, associando a chave pública do grupo com a sua identidade. Esses são chamados de *certificados dos grupos*. Assim, se os membros de um dado grupo IG_α acreditam que a chave pública PK_β pertence ao grupo IG_β, eles podem emitir um certificado $C_{SK_\alpha}^{IG_\beta}$, assinado com a chave privada SK_α do grupo IG_α, associando a identidade do grupo IG_β com a chave pública PK_β. Inicialmente, esse certificado é armazenado por todos os membro de grupo IG_α e do grupo IG_β. Além disso, os nós trocam periodicamente os seus certificados com os seus vizinhos, por meio de um mecanismo de troca de certificados. Como mais de um nó pode participar dos dois grupos, caso um membro do grupo IG_β receba dois ou mais certificados iguais, ele armazena apenas um dos certificados em seu repositório local.

Para armazenar os seus certificados, cada nó x_i possui dois repositórios locais: um repositório de certificados de grupos atualizados (G_i) e um repositório de certificados de grupos não-atualizados (G_i^N). Da mesma forma que no PGP-Like [9], descrito no Capítulo 2.1, os repositórios atualizados contém um subconjunto de certificados que são considerados válidos. Porém, diferente do PGP-*Like*, no SG-PKM o nó não envia mensagens solicitando a atualização desses certificados quando eles expiram, mas os armazena em seu repositório local de certificados de grupos não-atualizados. Esses certificados não-atualizados devem ser validados reativamente somente quando eles precisam ser utilizados.

Se um dado nó x_u precisa realizar uma comunicação segura com um outro nó x_v, e precisar certificar-se de que a chave pública pk_v pertecente ao nó x_v, ele precisa verificar a autenticidade dessa chave. Assim, quando um dado nó x_u, pertencente ao grupo IG_α, deseja autenticar a chave pública de um nó x_v, pertencente ao grupo IG_γ, ele obtém o certificado do nó x_v com a identificação do grupo ao qual esse pertence. Caso o nó x_v participe de mais grupos, ele pode apresentar ao nó x_u todos os certificados emitidos para

ele. Para autenticar o certificado apresentado, o nó x_u utiliza a chave pública do grupo ao qual pertence o nó x_v. No entanto, ele precisa autenticar a chave pública desse grupo (PK_γ). Para isso, ele deve obter no mínimo duas cadeias disjuntas de certificados de grupos válidos em seu repositório local, de forma similar ao apresentado em [9]:

a) o primeiro certificado da cadeia deve ser verificado diretamente pelo nó x_v usando a chave pública do seu grupo (PK_α);

b) os demais certificados devem ser verificados usando a chave pública contida no certificado anterior na cadeia;

c) por fim, o último certificado contém a chave pública do grupo IG_γ.

Depois que o nó x_u certifica-se de que a chave pública do nó x_v é válida, ambos os nós (x_u e x_v) podem criar uma chave uma chave de sessão para a realização da comunicação segura. Note que os nós x_u e x_v não precisam ser vizinhos físicos para realizarem uma autenticação mútua. Dessa forma, eles necessitam dos serviços de roteamento fornecidos pela rede.

3.2 Entrada de nós na rede e criação dos grupos

Para participar da rede, um dado nó x_u não precisa realizar nenhuma operação adicional. Caso o SG-PKM esteja sendo empregado em uma rede aberta, qualquer nó pode entrar na rede em qualquer momento. Por outro lado, caso o SG-PKM esteja sendo utilizado em uma rede fechada, ele precisa respeitar às regras dessa rede para poder participar de suas operações.

Contudo, caso esse nó x_u deseje fornecer ou soliticitar algum serviço que necessite do sistema de gerenciamento de chaves, é necessário que ele também faça parte desse sistema. Assim, quando um dado nó x_u desejar participar do SG-PKM, ele deve criar localmente o seu par de chaves pública e privada, denotados respectivamente como pk_u e sk_u. Em seguida, ele deve encontrar outros $m - 1$ nós que ele confia. Esse conjunto de m nós, incluindo x_u, formarão um grupo IG_α. Esses m nós precisam confiar uns nos outros e

essa confiabilidade segue a relação de amizade existente entre os usuários. Note que essa operação não precisa ser realizada simultaneamente por todos os m membros do grupo.

Os nós de um dado grupo IG_α trocam suas chaves públicas via um canal paralelo seguro, como em [9, 11]. Além disso, cada grupo IG_α deve possuir um par de chaves pública (PK_α) e privada (SK_α). A criação dessas chaves será detalhada na Seção 3.3. Antes de participar de um grupo, um dado nó x_u pode participar de todas operações da rede, desde que essas operações não necessitem dos serviços de gerência de chaves.

Assume-se, também, que a rede possua um mecanismo para identificação única dos nós. Dessa forma, cada nó deve possuir um endereço ou identificador. Com isso, cada grupo IG_α pode ser identificado usando um *hash* sobre a concatenação dos identificadores dos seus membros, i.e $IG_\alpha = \mathcal{H}(x_1 \| x_2 \| ... \| x_m)$. No entanto, qualquer outro mecanismo pode ser utilizado para identificar um grupo, desde que garanta a exclusividade da identificação.

3.3 Formação e distribuição colaborativa das chaves pública e privada de um grupo

No SG-PKM, para a construção e distribuição segura das chaves pública e privada de um grupo, assume-se a existência de um esquema de acordo de chaves sem a necessidade de uma entidade central [50, 14, 31]. Embora este trabalho tenha utilizado o esquema esquema de Pedersen [50], qualquer outro esquema de acordo de chaves sem uma Terceira Entidade Confiável (*Third Trusted Party* (TTP)) pode ser utilizado. As operações a seguir, de criação das chaves pública e privada de um grupo, são baseadas no esquema de Pedersen.

Na criação da chave pública do grupo, cada nó x_i escolhe aleatoriamente um segredo v_i pertencente à $\mathbb{Z}_q$ e calcula $PK_{\alpha_i} = g^{v_i}$, em que:

- q e p são dois números primos grandes, tal que q divide $p - 1$;
- $\mathbb{Z}_q$ é o conjunto de números inteiros positivos menores que q;
- g é um gerador de G_q;

- G_q é o subgrupo único de $\mathbb{Z}_p^*$ de ordem q.

Em seguida, cada nó x_i envia PK_{α_i} para todos os demais nós do grupo. Como os membros de um grupo não precisam ser vizinhos físicos, eles necessitam do roteamento para realizarem suas operações. Quando um nó x_j receber as $m-1$ partes, ele pode calcular a chave pública $PK_\alpha = \prod_{i=1}^{m} PK_{\alpha_i}$. Como todos os nós do grupo irão, eventualmente, receber as mesmas partes, todos conhecerão a mesma chave pública. Essa operação não precisa ser realizada ao mesmo tempo por todos os m membros. Assim, é possível que em um dado tempo t_0 alguns nós já possuam a chave pública PK_α, enquanto os demais membros apenas consigam formar PK_α em um tempo $t_x > t_0$.

Na construção da chave privada do grupo, cada nó x_i escolhe aleatoriamente uma função polinomial $f_i(z)$ sobre $\mathbb{Z}_q$ de grau $(t-1)$, de forma que $f_i(0) = v_i$. Em seguida, cada nó x_i calcula uma subparte $s_{ij} = f_i(j)$ para cada nó x_j, em que $j = 1, 2, .., m$, e envia essa subparte s_{ij} para o nó x_j. Após receber $(m-1)$ subpartes, o nó x_j pode calcular a sua parte da chave privada como $SK_{\alpha_j} = \sum_{i=1}^{n} s_{ij}$. Como na formação da chave pública PK_α, não é necessário que todos os nós calculem a sua subparte da chave privada SK_α ao mesmo tempo.

Essas chaves são geradas de forma que a chave pública de um grupo seja conhecida por todos os membros desse grupo e a chave privada seja compartilhada por todos os membros em um esquema de limiar (t, m) [56]. Nesse caso, cada nó possui uma parte da chave privada (SK_α) e essa chave somente pode ser reconstruída com as informações de t nós. Toda operação realizada por um grupo, como a emissão de um certificado ou assinatura de uma mensagem de grupo, precisa da participação de no minimo t membros desse grupo. Qualquer operação conjunta de $t-1$ nós não é capaz de reconstruir essa chave. Dessa forma, mesmo que até $m-t$ nós de um grupo saiam da rede, ainda é possível que os demais membros do grupo realizem operações em nome do grupo, como emissão, atualização ou revogação de certificados.

O uso da criptografia de limiar visa aumentar a disponibilidade e a resistência a ataques, pois não é necessário que todos os membros do grupo participem da reconstrução da chave privada do grupo. Além disso, caso mais que t nós participem da operação, o

resultado será o mesmo, e a chave privada poderá ser reconstruída. Além disso, t deve ser no mínimo $\lfloor m/2 \rfloor + 1$, para que o esquema mantenha a consistência nas decisões [56].

A geração distribuída das chaves de um grupo IG_α e o uso de um esquema de limiar melhoram a tolerância a ataques no SG-PKM [54, 55]. Além disso, a formação de grupos baseada nas relações de amizade dos usuários visa diminuir a probabilidade de identidades falsas no sistema.

3.4 Emissão e distribuição dos certificados de nós

Após a formação do grupo IG_α e a criação colaborativa de seu par de chaves pública e privada, os nós membros desse grupo emitem os certificados de chave pública entre si, assinados com a chave privada do grupo (SK_α). Assim, qualquer nó da rede pode verificar a autenticidade desse certificado conhecendo a chave pública PK_α do grupo IG_α. Todos os nós pertencentes ao grupo receberão um certificado assinado com a chave privada desse grupo, associando a sua identidade com a sua chave pública.

Um certificado, emitido por no mínimo t membros do grupo IG_α ao nó x_i, é representado como segue:

$$C_{SK_\alpha}^{x_i} = ((T_{validade} \| x_i \| pk_i \| MAC(IG_\alpha))_{SK_\alpha} \| IG_\alpha) \tag{3.1}$$

no qual:

- $T_{validade}$ é o tempo de validade do certificado;
- x_i é a identidade do nó associado ao certificado;
- pk_i é a chave pública do nó x_i;
- $MAC(IG_\alpha)$ é um código de autenticação de mensagem da identidade do grupo IG_α;
- SK_α é a chave privada do grupo IG_α utilizada para assinar o certificado;
- IG_α é a identidade de grupo emissor do certificado.

3.5 Emissão e distribuição dos certificados de grupos

A chave pública PK_β de um dado grupo IG_β também precisa ser certificada, para que essa chave possa ser utilizada por outros nós para a validação dos certificados assinados com a chave privada SK_β. Para isso, os grupos podem emitir certificados entre eles associando a chave pública de um dado grupo com a sua identidade. Em outras palavras, os membros do grupo IG_α podem emitir um certificado $C_{SK_\alpha}^{IG_\beta}$ para o grupo IG_β, se os membros do grupo IG_α acreditam que a chave pública PK_β pertence ao grupo IG_β.

Podem existir várias razões para que os membros do grupo IG_α acreditem que a chave pública PK_β pertence ao grupo IG_β, entre elas:

a) no mínimo t nós do grupo IG_α podem confiar em dois ou mais nós pertencentes ao grupo IG_β, e esses nós lhe forneceram a chave pública do grupo IG_β;

b) dois ou mais nós do grupo IG_β também fazem parte do grupo IG_α.

A exigência de no mínimo dois nós relacionando os dois grupos visa melhorar a confiabilidade na avaliação da chave pública do grupo IG_β. Um certificado emitido pelos membros de grupo IG_α ao grupo IG_β é representado como segue:

$$C_{SK_\alpha}^{IG_\beta} = ((T_{validade} \| IG_\beta \| MAC(IG_\alpha) \| PK_\beta)_{SK_\alpha} \| IG_\alpha) \tag{3.2}$$

no qual:

- $T_{validade}$ é o tempo de expiração do certificado $C_{SK_\alpha}^{IG_\beta}$;
- IG_β é a identidade do grupo associado ao certificado;
- IG_α é a identidade do grupo emissor do certificado;
- $MAC(IG_\alpha)$ é o código de autenticação de mensagem da identidade do grupo IG_α;
- PK_β é a chave pública do grupo IG_β;
- SK_α é a chave privada do grupo IG_α utilizada para assinar o certificado $C_{SK_\alpha}^{IG_\beta}$.

3.6 Trocas dos certificados de grupos

O mecanismo de troca de certificados consiste na troca periódica dos certificados de grupos entre nós fisicamente vizinhos. Inicialmente, cada nó possui em seus repositórios locais, apenas os certificados dos grupos aos quais ele pertence e os certificados que os nós dos seus grupos emitiram para outros grupos. Com a troca periódica de certificados com os seus vizinhos, cada nó aumenta a quantidade de certificados em seus repositórios locais.

Cada nó envia periodicamente uma mensagem solicitando uma troca de certificados com os seus vizinhos. Essa mensagem pode ser enviada juntamente (*piggybacking*) com as mensagens usadas pelo protocolo de controle de acesso ao meio para descoberta dos nós vizinhos. Esse mecanismo de troca de certificados, apresentado no Algoritmo 3.1, funciona da seguinte forma:

a) um dado nó x_i envia aos seus vizinhos um *hash* dos seus repositórios locais, e solicita a eles que enviem os certificados que ele ainda não possui;

b) cada vizinho responde com uma mensagem contendo os certificados que ele possui em seus repositórios locais e que não estão armazenados nos repositórios do nó x_i;

c) ao receber os certificados dos seus vizinhos, o nó x_i armazena esses certificados em seu repositório local de certificados de grupos não-atualizados (G_i^N).

Algoritmo 3.1 EXCHANGE()

nó x_i ***a cada*** T_{troca} ***segundos.***

1: $msg.type \leftarrow$ getCertificates
2: $msg.data \leftarrow hash(G_i)$
3: $send(msg, N(x_i))$
4: **para todo** resposta $msg.type = sendCertificates$ **de** x_j **faça**
5: $\quad G_i^N \leftarrow G_i^N + msg.certificates$
6: **fim para**

nó $x_j \in N(x_i)$ ***ao receber*** $msg.type = getCertificate$

1: $msg.type \leftarrow$ sendCertificate
2: $msg.certificate \leftarrow cert \in G_j \cup G_j^N : cert \notin msg.data$
3: $send(msg, x_i)$

As trocas de certificados são realizadas em intervalos de tempo, denominados T_{troca}. Cada nó tem uma frequência para a realização das trocas de certificados com os nós vizinhos.

Com o mecanismo de troca de certificados, os nós adquirem mais informações sobre os certificados que foram emitidos. Com essas informações, eles aumentam o tamanho dos seus repositórios de certificados não-atualizados. Caso eles desejem, eles podem fazer a validação dos certificados (Seção 3.8).

3.7 Autenticação das chaves públicas

Quando um nó x_i deseja autenticar a chave pública pk_j de um nó x_j, ele solicita ao nó x_j o certificado emitido para a sua chave pública. Como x_j pode participar de vários grupos, ele pode apresentar qualquer um dos certificados que foram emitidos a ele. Dessa forma, x_i pode escolher um ou mais certificados para validar. Dentro de cada certificado, o nó x_i pode saber a identificação do grupo emissor. O Algoritmo 3.2 demonstra esse caso, em que o nó x_i realiza a autenticação de certificado $C_{SK_\gamma}^{x_j}$, assinado pelos membros do grupo IG_γ.

Caso o nó x_i escolha o certificado $C_{SK_\gamma}^{x_j}$ para autenticação, ele precisa utilizar a chave pública PK_γ para validar esse certificado. Embora a autenticação da chave pública pk_j seja realizada de forma simples, o nó x_i precisa confirmar que a chave pública (PK_γ) do grupo IG_γ é uma chave correta. Assim, antes de usar PK_γ, x_i precisa autenticar essa chave. Essa autenticação é realizada via cadeias de certificados de grupos. Para isso, o nó x_i procura no mínimo duas cadeias disjuntas de certificados ligando os grupos que ele pertence ao grupo IG_γ em seu repositório local de certificados de grupos atualizados. Caso $\exists(PK_\alpha \rightrightarrows PG_\gamma) \in G_i : x_i \in IG_\alpha$, ele pode validar a chave pública PK_γ do grupo IG_γ e, em seguida, validar o certificado $C_{SK_\gamma}^{x_j}$ do nó x_i.

Porém, se $\nexists(PK_\alpha \rightrightarrows PK_\gamma) \in G_i : x_i \in IG_\alpha$, ele juntará o seu repositório de certificados de grupos atualizados com o repositório de certificados de grupos atualizados do nó x_j ($G_1 = G_i \cup G_j$). Então, novamente o nó x_i procura no mínimo duas cadeias de certificados de grupos. Da mesma forma, caso $\exists(PK_\alpha \rightrightarrows PK_\gamma) \in G_1 : x_i \in IG_\alpha$, o nó x_i

Algoritmo 3.2 AUTENTICATION($C_{SK_\gamma}^{x_j}$)

1: $IG_{x_i} \leftarrow$ Grupos de x_i
2: $Counter_{i \to j} \leftarrow 0$
3: **para todo** $IG_i \in IG_{x_i}$ **faça**
4: **para todo** $(IG_i \rightsquigarrow IG_\gamma) \in G_i$ **faça**
5: $Counter_{i \to j} = Counter_{i \to j} + 1$
6: **fim para**
7: **se** $Counter_{i \to j} < 2$ **então**
8: **para todo** $(IG_i \rightsquigarrow IG_\gamma) \in G_i \cup G_j$ **faça**
9: $Counter_{i \to j} = Counter_{i \to j} + 1$
10: **fim para**
11: **fim se**
12: **se** $Counter_{i \to j} < 2$ **então**
13: **para todo** $(IG_i \rightsquigarrow IG_\gamma) \in G_i \cup G_i^N$ **faça**
14: $validate(C_{SK_\alpha}^{IG_\beta})\ \forall\ C_{SK_\alpha}^{IG_\beta}$ não atualizado na cadeia.
15: **se** todos $C_{SK_\alpha}^{IG_\beta}$ estão validados **então**
16: $Counter_{i \to j} = Counter_{i \to j} + 1$
17: **fim se**
18: **fim para**
19: **fim se**
20: **fim para**
21: **se** $Counter_{i \to j} \geq 2$ **então**
22: **retorna verdadeiro**
23: **senão**
24: **retorna falso**
25: **fim se**

pode validar a chave pública PK_γ do grupo IG_γ e autenticar o nó x_j.

Se mesmo após a junção dos repositórios, $\nexists(PK_\alpha \rightrightarrows PK_\gamma) \in G_1 : x_i \in IG_\alpha$, o nó x_i criará $G_2 = G_i \cup G_i^N$ e tentará encontrar as duas cadeias disjuntas em G_2. Caso $\exists(PK_\alpha \rightrightarrows PK_\gamma) \in G_2 : x_i \in IG_\alpha$, x_i precisa verificar se as associações representadas pelos certificados não-atualizados ainda são válidas. A validação de um certificado de grupo é detalhada na Seção 3.8. Caso ele encontre as duas cadeias de certificados de grupo validados e atualizados, ele pode validar a chave pública PK_γ do grupo IG_γ e autenticar o nó x_j.

Por fim, caso $\nexists(PK_\alpha \rightrightarrows PK_\gamma) \in G_2 : x_i \in IG_\alpha$, ele não será capaz de autenticar a chave pública do grupo IG_γ e, como consequência, não será capaz de autenticar o nó x_j.

3.8 Validação dos certificados de grupo

Relembrando, os certificados recebidos no mecanismo de trocas de certificados são armazedos nos repositórios de certificados não-atualizados dos nós. Além disso, todos os certificados com o tempo de vida expirado, também são armazenados nesse repositório. Quando um nó x_i deseja validar um certificado armazenado em seu repositório de certificados não-atualizados, ele precisa enviar uma mensagem ao membros do grupo emissor desse certificado. A validação deve ser feita por no mínimo t nós pertencentes ao grupo que emitiu o certificado.

O Algoritmo 3.3 demonstra o processo de validação, em que o nó x_i envia uma mensagem de Pedido de Validação (VREQ) para todos os membros do grupo emissor do certificado $C_{SK_\alpha}^{IG_\beta}$, e aguarda pelo menos t mensagens assinadas, válidas e positivas, de Resposta de Validação (VREP).

Algoritmo 3.3 VALIDATE($C_{SK_\alpha}^{IG_\beta}$)

no nó solicitante x_i***:***

1: $Reply\_Counter_{C_{SK_\alpha}^{IG_\beta}} \leftarrow 0$
2: $msg.tipo \leftarrow VREQ$
3: $msg.certificado \leftarrow C_{SK_\alpha}^{IG_\beta}$
4: **para todo** $x_v \in IG_\alpha$ **faça**
5: $\quad send(msg, x_v)$
6: **fim para**

x_i ***recebendo um*** $VREP$ **verdadeiro*:***

1: $C_{SK_\alpha}^{IG_\beta} \leftarrow msg.certificado$
2: $Reply\_Counter_{C_{SK_\alpha}^{IG_\beta}} \leftarrow Reply\_Counter_{C_{SK_\alpha}^{IG_\beta}} + 1$
3: **se** $Reply\_Counter_{C_{SK_\alpha}^{IG_\beta}} \geq t$ **então**
4: $\quad$ **retorna verdadeiro**
5: **fim se**

nó $x_v \in IG_\alpha$ ***recebendo um*** $VREQ$***:***

1: $msg.type \leftarrow VREP$
2: **se** $msg.certificate$ ainda válido **então**
3: $\quad reply(msg, \textbf{verdadeiro})$
4: **senão**
5: $\quad reply(msg, \textbf{falso})$
6: **fim se**

3.9 Atualização dos certificados

Antes que o tempo de validade de um certificado expire, o seu grupo emissor pode emitir uma versão atualizada desse mesmo certificado, com um novo tempo de expiração. Um subconjunto de t nós pertencentes a um grupo podem decidir emitir um certificado atualizado se eles acreditam que a associação 'nó - chave pública' contida nesse certificado ainda esteja correta. A forma de atualização dos certificados de nós e dos certificados de grupo são diferentes. Em todos os casos, é necessário um subconjunto de t nós do grupo emissor para realizar a reemissão do certificado. O uso de t nós em vez dos m do grupo minimiza a sobrecarga na rede física.

3.9.1 Certificados de nós

A atualização de um certificado de nó é iniciada pelo próprio nó, que solicita aos demais membros do grupo a atualização do seu certificado. Se um subconjunto t de um dado grupo IG_γ não possui qualquer motivo para não acreditar na associação entre a identidade do nó x_i e a chave pública pk_i em um dado certificado de nó $C_{SK_\gamma}^{x_i}$, eles podem emitir uma atualização desse certificado, com um novo tempo de validade. Eles fazem isso enviando ao próprio nó x_i uma mensagem de atualização do certificado, assinada com suas respectivas subpartes da chave privada do grupo IG_γ. Quando o nó x_i receber t mensagens de atualização do seu certificado, ele se encarrega de enviar uma cópia do certificado atualizado para os demais membros do grupo IG_γ. O Algoritmo 3.4 demonstra o processo de atualização do certificado $C_{SK_\gamma}^{x_i}$ do nó x_i, assinado pelos membros do grupo IG_γ.

3.9.2 Certificados de grupos

Os certificados de grupos também podem ser atualizados por um subconjunto de t nós do grupo que originalmente emitiu o certificado. Nesse caso, caso um nó x_i, membro do grupo IG_β precise que o certificado $C_{SK_\alpha}^{IG_\beta}$ seja atualizado, ele pode pode solicitar a atualização desse certificado aos membros do grupo IG_α.

Algoritmo 3.4 UPDATE($C^{x_i}_{SK_\gamma}$)

no x_i ***desejando atualizar o seu certificado*** $C^{x_i}_{SK_\gamma}$

1: $msg.type \leftarrow$ requestRenewing
2: $msg.certificate \leftarrow C^{x_i}_{SK_\gamma}$
3: $send(msg, IG_\gamma \setminus x_i)$
4: $Update\_Counter_{C^{x_i}_{SK_\gamma}} \leftarrow 1$

nó $x_j \in IG_\gamma$ ***ao receber um pedido de atualização do nó*** x_i

1: **se** x_j acredita na associação do nó x_i com a chave pública pk_i **então**
2: $msg.type \leftarrow$ nodeRenewing
3: $msg.certificate \leftarrow C^{x_i}_{SK_\gamma}$
4: $send(msg, x_i)$
5: **fim se**

nó x_i ***ao receber uma mensagem de atualização de*** $C^i_{SK_\gamma}$

1: $Update\_Counter_{C^{x_i}_{SK_\gamma}} \leftarrow Update\_Counter_{C^{x_i}_{SK_\gamma}} + 1$
2: **se** $Update\_Counter_{C^{x_i}_{SK_\gamma}} \geq t$ **então**
3: **atualizar** $C^{x_i}_{SK_\gamma}$
4: $msg.type \leftarrow$ nodeRenewed
5: $msg.certificate \leftarrow C^{x_i}_{SK_\gamma}$
6: $send(msg, IG_\gamma \setminus x_i)$
7: **fim se**

Para isso, o nó x_i envia um pedido de atualização do certificado para todo os membros do grupo IG_α, e aguarda por no mínimo t respostas de atualização de certificado, cada uma assinada com uma subparte distinta da chave privada do grupo IG_α. Além disso, ao enviar a sua subparte da atualização do certificado ao nó x_i, cada nó envia também uma lista dos outros nós do sistema que solicitaram uma validação do certificado que está sendo atualizado.

A nova versão do certificado, com o novo tempo de validade, é enviada, pelo nó x_i, para todos os nós do grupo emissor, para todos os nós do grupo que teve o seu certificado atualizado e para todos os nós que solicitaram uma atualização desse certificado anteriormente. Com o objetivo de minimizar a sobrecarga de comunicação, o nó x_i pode enviar o certificado atualizado apenas para os nós que solicitaram uma validação desse certificado mais recentemente. Se um dado nó x_v não receber uma versão atualizado de um certificado após a sua expiração, esse certificado será movido para o repositório não atualizado do nó x_v, e esse certificado deve ser atualizado, reativamente, quando ele precisar ser utilizado novamente. O Algoritmo 3.5 apresenta o procedimento para

renovação de um dado certificado de grupo $C_{SK_\alpha}^{IG_\beta}$.

Algoritmo 3.5 UPDATE($C_{SK_\alpha}^{IG_\beta}$)

no x_i ***desejando atualizar o seu certificado*** $C_{SK_\alpha}^{IG_\beta}$

1: $msg.type \leftarrow$ requestRenewing
2: $msg.certificate \leftarrow C_{SK_\alpha}^{IG_\beta}$
3: $send(msg, IG_\alpha \setminus x_i)$
4: $Update\_Counter_{C_{SK_\alpha}^{IG_\beta}} \leftarrow 1$

nó $x_j \in IG_\alpha$ ***ao receber um pedido de atualização do nó*** x_i

1: **se** x_j acredita na associação do grupo IG_β com a chave pública PK_β **então**
2: $msg.type \leftarrow$ groupRenewing
3: $msg.certificate \leftarrow C_{SK_\alpha}^{IG_\beta}$
4: $msg.listOfNodes \leftarrow$ Lista de nós que solicitaram validação do certificado
5: $send(msg, x_i)$
6: **fim se**

nó x_i ***ao receber uma mensagem de atualização de*** $C_{SK_\alpha}^{IG_\beta}$

1: $Update\_Counter_{C_{SK_\alpha}^{IG_\beta}} \leftarrow Update\_Counter_{C_{SK_\alpha}^{IG_\beta}} + 1$
2: $L \leftarrow msg.listOfNodes$
3: **se** $Update\_Counter_{C_{SK_\alpha}^{IG_\beta}} \geq t$ **então**
4: **atualizar** $C_{SK_\alpha}^{IG_\beta}$
5: $msg.type \leftarrow$ nodeRenewed
6: $msg.certificate \leftarrow C_{SK_\alpha}^{IG_\beta}$
7: $send(msg, IG_\gamma \setminus x_i)$
8: $send(msg, L)$
9: **fim se**

3.10 Revogação dos certificados

No SG-PKM, tanto os certificados de nós como os certificados de grupos podem ser revogados. Esses dois tipos de certificados são emitidos por membros de um grupo e são assinados com a chave pública desse grupo. Dessa forma, se no mínimo t membros acreditarem que um outro nó ou grupo está comprometido, eles podem revogar o certificado emitido.

Um certificado de nó ou grupo pode ser revogado implicitamente ou explicitamente. A revogação implícita desses certificados é baseada nos seus respectivos tempos de validade. Quando um dado certificado tem o seu tempo de validade expirado, e ele não foi atualizado pelo seu grupo emissor, então ele é considerado um certificado revogado. A

revogação implícita acontece automática e localmente para todos os certificados armazenados no repositório local de certificados atualizados de cada nó. A revogação explítica dos certificados será explicada nas Seções 3.10.1 e 3.10.2.

Além disso, um dado nó pode revogar a sua própria chave pública quando acreditar que a sua chave privada está comprometida, ou ainda, os membros de um grupo podem revogar a chave pública desse grupo se acreditarem que a chave privada está comprometida. A auto-revogação será discutida na Seção 3.10.3

3.10.1 Revogação dos certificados de nós

Muitas razões podem levar um certificado a se tornar inválido antes do seu tempo de expiração. Alguns exemplos dessas razões são [18]:

a) mudanças no relacionamento entre o emissor do certificado e o proprietário da chave (ex. dois usuário não possuem mais um relacionamento de amizade);

b) a suspeita que a chave privada associada com o certificado foi comprometida.

Sob essas circunstâncias, o grupo emissor do certificado pode desejar revogar explicitamente um certificado. Na revogação explícita, os membros de um dado grupo IG_α podem revogar o certificado de um outro nó pertencente a esse grupo, i.e. o certificado $C_{SK_\alpha}^{x_i}$ pertecente ao nó x_i. É necessário que pelo menos t membros do grupo IG_α concordem com a revogação do certificado.

O Algoritmo 3.6 apresenta o processo de revogação explícita de um certificado de nó. No exemplo, um dado nó x_j, membro de um grupo IG_α, deseja revogar o certificado $C_{SK_\alpha}^{x_i}$ do nó x_i, assinado com a chave pública SK_α do grupo IG_α. Nesse caso, ele envia um pedido de revogação para os demais membros do grupo emissor (IG_α). Cada membro do grupo IG_α, ao receber o pedido de revogação do certificado, toma uma decisão com base no seu conhecimento sobre o nó x_i, baseado nas informações dos níveis de prevenção e reputação sobre esse nó. Caso ele também tenha motivos para revogar o certificado, ele retorna uma mensagem ao nó x_j, assinada com a sua subparte da chave privada do grupo IG_α, aceitando o pedido de revogação do certificado $C_{SK_\alpha}^{x_i}$.

Algoritmo 3.6 REVOKE($C^{x_i}_{SK_\alpha}$)

no x_j ***desejando revogar o certificado*** $C^{x_i}_{SK_\alpha}$

1: $msg.type \leftarrow$ requestRevocation
2: $msg.certificate \leftarrow C^{x_i}_{SK_\alpha}$
3: $send(msg, IG_\alpha \setminus x_i)$
4: $Revoke\_Counter_{C^{x_i}_{SK_\alpha}} \leftarrow 1$

nó $x_j \in IG_\alpha$ ***ao receber um pedido de revogação do nó*** x_i

1: **se** x_j também deseja revogar o certificado $C^{x_i}_{SK_\alpha}$ **então**
2: $msg.type \leftarrow$ nodeRevocation
3: $msg.certificate \leftarrow C^{x_i}_{SK_\alpha}$
4: $send(msg, x_i)$
5: **fim se**

nó x_i ***ao receber uma aceitação de revogação de*** $C^{x_i}_{SK_\alpha}$

1: $Update\_Counter_{C^{x_i}_{SK_\alpha}} \leftarrow Update\_Counter_{C^{x_i}_{SK_\alpha}} + 1$
2: **se** $Update\_Counter_{C^{x_i}_{SK_\alpha}} \geq t$ **então**
3: **revogar** $C^{x_i}_{SK_\alpha}$
4: $CRL_i \leftarrow CRL_i \cup C^{x_i}_{SK_\alpha}$
5: $msg.type \leftarrow$ nodeRevocated
6: $msg.certificate \leftarrow C^{x_i}_{SK_\alpha}$
7: $send(msg, IG_\gamma \setminus x_i)$
8: **para todo** IG que tenha emitido um certificado ao grupo IG_α **faça**
9: $send(msg, IG \setminus x_i)$
10: **fim para**
11: **fim se**

qualquer nó x_v ***ao receber uma mensagem para revogar*** $C^{x_i}_{SK_\alpha}$

1: $CRL_v \leftarrow CRL_v \cup C^{x_i}_{SK_\alpha}$
2: **se** x_v é membro de um grupo emissor de um certificado para o grupo IG_α **então**
3: $L \leftarrow$ Lista de nós que solicitaram uma validação do certificado de grupo IG_α
4: $send(msg, L)$
5: **fim se**

Caso o nó x_j obtenha no mínimo t mensagens de aceitação do pedido de revogação do certificado, esse certificado é considerado revogado. Em seguida, o nó x_j deve enviar a mensagem completa de revogação do certificado $C^{x_i}_{SK_\alpha}$, agora assinada com a chave privada do grupo IG_α, para todos os demais membros do grupo. Além disso, ele deve enviar essa mensagem para todos os membros dos grupos que emitiram um certificado para o grupo IG_α. Esses grupos devem, então, propagar essa informação aos demais nós que solicitaram a validação ou atualização do certificado do grupo que possui o nó comprometido. Como isso, todos os nós que possuem o certificado da chave pública (PK_α) do grupo IG_α serão informados que esse grupo posusui um certificado revogado.

Essa última etapa é importante pois, como foi detalhado na Seção 3.7, quando um nó desejar validar a autenticidade do certificado $C^{x_i}_{SK_\alpha}$, ele precisa utilizar a chave pública do grupo IG_α (PK_α). Para isso, ele deve realizar a validação da chave pública do grupo IG_α juntamente com os grupos que emitiram um certificado associando a chave pública PK_α à identidade do grupo IG_α. Dessa forma, os membros dos grupos que emitiram um certificado para IG_α devem sempre informar da existência de algum nó que possui o seu certificado comprometido nesse grupo.

Todos os nós que receberem uma mensagem de revogação de um certificado, irão armazenar essa informação em uma Lista de Certificados Revogados (*Certificate Revocation List* (CRL)) local, e usarão essa informação antes de autenticarem ou darem informações sobre o dado certificado. Tantos os certificados revogados de nós como os de grupos, que serão apresentados na próxima Seção, são armazenados em uma mesma lista. A CRL facilita o processo de autenticação, minimizando os custos computacionais na pesquisa de certificados válidos.

3.10.2 Revogação dos certificados de grupos

Da mesma forma que na revogação dos certificados de nós, um certificado de grupo pode ser revogado implicitamente ou explicitamente. Nesse caso, a revogação implítica de um certificado de grupo também é baseada no tempo de validade do certificado e não é necessária nenhuma intervenção da ICP nessa operação. Os certificados expirados são armazenados nos repositório locais de certificados não-atualizados dos nós.

O Algoritmo 3.7 apresenta um dado nó x_i revogando explicitamente um dado certificado de grupo $C^{IG_\beta}_{SK_\alpha}$, pertencente ao grupo IG_β. Nesse caso, o nó x_i cria uma mensagem msg do tipo $requestRevogacation$, informando o certificado a ser revogado, e envia para todos os demais membros do grupo IG_α. Como na revogação de um certificado de nó, um membro do grupo IG_α, ao receber o pedido de revogação do certificado de grupo, pode aceitar esse pedido, e retornar uma mensagem de aceitação da revogaçao do certificado de grupo. Essa mensagem deve ser assinada com a sua respectiva subparte da chave privada do grupo IG_α. Essa mensagem também devem conter a lista dos nós que solicitaram uma

validação do certificado que está sendo revogado ($C_{SK_\alpha}^{IG_\beta}$).

Algoritmo 3.7 REVOKE($C_{SK_\alpha}^{IG_\beta}$)

no x_i ***desejando revogar o certificado*** $C_{SK_\alpha}^{IG_\beta}$

1: $msg.type \leftarrow$ requestRevocation
2: $msg.certificate \leftarrow C_{SK_\alpha}^{IG_\beta}$
3: $send(msg, IG_\alpha \setminus x_i)$
4: $Revoke\_Counter_{C_{SK_\alpha}^{IG_\beta}} \leftarrow 1$

nó $x_j \in IG_\alpha$ ***ao receber um pedido de revogação do grupo*** IG_β

1: **se** x_j também deseja revogar o certificado $C_{SK_\alpha}^{IG_\beta}$ **então**
2: $msg.type \leftarrow$ groupRevocation
3: $msg.listOfNodes \leftarrow$ Lista de nós que solicitaram validação de $C_{SK_\alpha}^{IG_\beta}$
4: $msg.certificate \leftarrow C_{SK_\alpha}^{IG_\beta}$
5: $send(msg, x_i)$
6: **fim se**

nó x_i ***ao receber uma aceitação de revogação de*** $C_{SK_\alpha}^{IG_\beta}$

1: $Revoke\_Counter_{C_{SK_\alpha}^{IG_\beta}} \leftarrow Revoke\_Counter_{C_{SK_\alpha}^{IG_\beta}} + 1$
2: $L \leftarrow msg.listOfNodes$
3: **se** $Revoke\_Counter_{C_{SK_\alpha}^{IG_\beta}} \geq t$ **então**
4: $CRL_i \leftarrow CRL_i \cup C_{SK_\alpha}^{IG_\beta}$
5: $G_i \leftarrow G_i \cap C_{SK_\alpha}^{IG_\beta}$
6: $G_i^N \leftarrow G_i^n \cup C_{SK_\alpha}^{IG_\beta}$
7: $msg.type \leftarrow$ nodeRevocated
8: $msg.certificate \leftarrow C_{SK_\alpha}^{i}$
9: $send(msg, L \cup IG_\alpha \setminus x_i)$
10: **fim se**

um nó x_v ***ao receber uma mensagem de revogação de*** $C_{SK_\alpha}^{IG_\beta}$

1: $CRL_v \leftarrow CRL_v \cup C_{SK_\alpha}^{IG_\beta}$
2: $G_v \leftarrow G_v \cap C_{SK_\alpha}^{IG_\beta}$
3: $G_v^N \leftarrow G_v^N \cup C_{SK_\alpha}^{IG_\beta}$

O nó x_i, ao receber no mínimo t respostas de aceitação do pedido de revogação do certificado de grupo, pode considerar o certificado revogado. Ele armazena esse certificado em sua CRL local e, em seguida, envia a mensagem de revogação (msg do tipo *groupRevocation*), agora assinada com a chave privada do grupo IG_α, para todos os membros do grupo IG_α e para todos os nós que solicitaram uma atualização de $C_{SK_\alpha}^{IG_\beta}$.

Ao receber uma mensagem msg do tipo *groupRevocation*, um nó move o certificado de grupo revogado do seu repositório local de certificados atualizados para o repositório

local de certificados não-atualizados e também armazena esse certificado em sua CRL local.

3.10.3 Auto-revogação

Caso um nó acredite que a sua própria chave privada está comprometida, ele pode informar os demais membros dos grupos que ele participa, para que esses realizem a revogação do certificado do nó comprometido. Dessa forma, suponha que um dado nó x_c, membro de um dado grupo IG_δ, descobre que a sua chave privada (sk_c) está comprometida, ele deve enviar uma mensagem de auto-revogação de sua chave pública aos demais membros do grupo IG_δ, para informar esses nós. Quando os membros do grupo IG_δ receberem essa mensagem, eles devem revogar explicitamente o certificado $C_{SK_\delta}^{x_c}$ do nó x_c.

Da mesma forma, um membro de um dado grupo IG_δ pode acreditar que os demais membros de seu grupo estão comprometidos e, consequentemente, a chave privada desse grupo também está comprometida. Nesse caso, ele pode enviar uma mensagem para os membros de todos os grupos que emitiram um certificado para o grupo IG_δ. Os membros desses grupos podem, então, revogar explicitamente o certificado que eles emitiram para o grupo IG_δ, quando receberem no mínimo t mensagens distintas de auto-revogação de um grupo.

3.11 Arquitetura de suporte ao novo esquema de gerenciamento de chaves

Tendo definido os objetivos, restrições e características de funcionamento do SG-PKM, a arquitetura SAMNAR [40], um acrônimo para *Survivable Ad hoc and Mesh Network ARchitecture*, foi escolhida para dar suporte ao novo esquema de gerenciamento de chaves. Essa arquitetura garante a sobrevivência do sistema por meio de uma cooperação adaptativa entre três linhas de defesa: preventiva, reativa e tolerância. Esta seção apresenta o funcionamento geral da arquitetura e como ela é aplicada no SG-PKM.

3.11.1 Funcionamento da arquitetura

Como ilustrado na Figura 3.3, a arquitetura *Survivable Ad hoc and Mesh Network ARchitecture* (SAMNAR) [40] possui três módulos distintos: **sobrevivência**, **comunicação** e **coleta**.

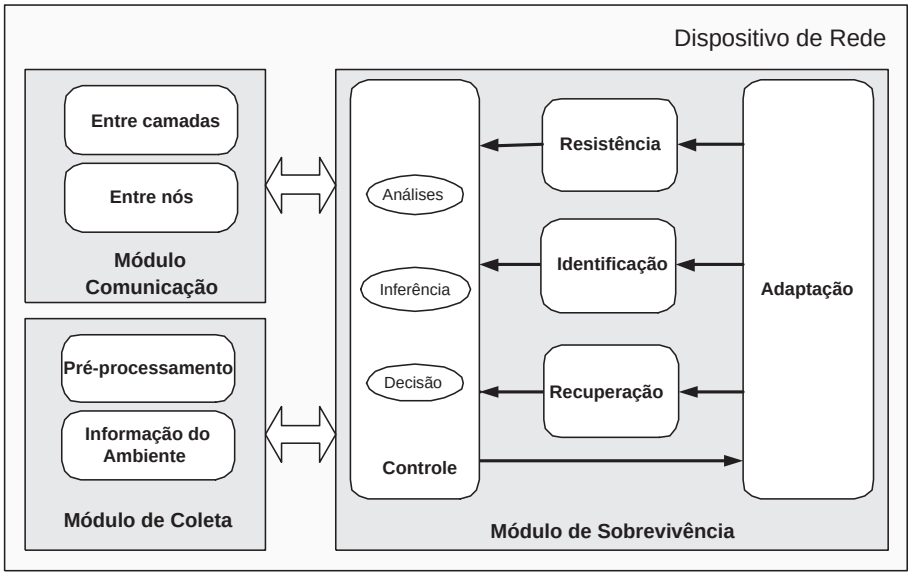


Figura 3.3: Arquitetura SAMNAR
Fonte: [40]

O **módulo de sobrevivência** gerencia cinco componentes, sendo eles: resistência, recuperação, identificação, adaptação e controle. O **componente de resistência** é composto por mecanismos locais de prevenção, tais como *firewalls* pessoais, antivírus, *anti-spyware*, operações criptográficas e outros.

O **componente de identificação** compreende mecanismos reativos para identificar comportamentos maliciosos, tais como sistemas de detecção de intrusos ou sistemas de reputação. O sistema de reputação, por exemplo, é responsável por coletar, distribuir e agregar uma avaliação sobre o comportamento dos nós participantes do sistema [53].

O **componente de recuperação** é responsável por fornecer mecanismos que aumentem a tolerância a ataques dos serviços essenciais da rede. Mecanismos para restaurar informações ou funcionalidades corrompidas, como replicação ou redundância podem ser empregados. Contudo, as estratégias de redundância devem considerar limitações de recursos.

O **componente de adaptação** é responsável por executar alterações baseadas nas análises, inferências e decisões do componente de controle. Essas alterações podem ser, por exemplo, nas regras do *firewall* pessoal, nos valores de limiares do sistema de reputação ou nos requisito de redundância das operações do sistema de gerência de chaves. Esse componente também é responsável por aprender com as ações anteriores e, posteriormente, tomar as mesmas decisões caso um nó ou a rede apresente condições similares.

Por fim, o **componente de controle** recebe informações dos módulos de comunicação e coleta, bem como dos componentes de resistência, identificação e recuperação, tais como informações do tamanho da chave usada nas operações criptográficas, o algoritmo criptográfico, a última data de atualização das bases de dados dos antivírus e *anti-spyware*, estatísticas sobre ataques e intrusões, entre outras. Ele correlaciona e analisa todas essas informações, e envia as suas decisões para o **componente de adaptação**.

O **módulo de comunicação** é responsável pelas comunicações entre camadas e entre nós. Nele, o **componente de comunicação entre camadas** fornece informações de diferentes camadas da rede para o componente de controle. Já o **componente de comunicação entre nós** fornece comunicação, troca e sincronização de informações entre os nós, visando garantir a sobrevivência de todas as configurações ou estatísticas sobre detecções de intrusão.

O **módulo de coleta** é responsável por obter todos os dados necessários pelo módulo de sobrevivência. Esse módulo é composto pelo **componente de pré-processamento**, responsável por tratar os dados antes que esses sejam enviados ao módulo de sobrevivência, e pelo **componente de informação do ambiente**, reponsável pelo armazenamento das informações sobre as condições da rede e, quando solicitado, enviar essas informações ao módulo de sobrevivência.

3.11.2 Aplicação no SG-PKM

Nem todos os módulos e componentes da arquitetura SAMNAR precisam ser implementados no SG-PKM. Alguns desses módulos ou componentes servem apenas como entradas para o funcionamento do esquema. O SG-PKM usa basicamente os compo-

nentes do módulo de sobrevivência, visto que esse é o principal objetivo do sistema. A Tabela 3.1 apresenta uma descrição de como os componentes do módulo de sobrevivência são aplicados no SG-PKM.

Tabela 3.1: Aplicação da arquitetura SAMNAR no SG-PKM

Componenente	**Aplicação no SG-PKM**
Resistência	• Operações criotográficas como: assinaturas digitais e Código de Autenticação de Mensagens (*Message Authentication Code* (MAC))
Identificação	• Entradas de um sistema de reputação
Recuperação	• Formação de grupos, que servem como testemunhas das trocas das chaves públicas • Exigência de cadeias de certificados disjuntas para a autenticação
Adaptação	• Alteração dos parâmetros de redundância e valores de limiares
Controle	• Analisa os valores recebidos dos outros componentes • Envia valores de parâmetros e limiares ao componente de adaptação

Os demais módulos da arquitetura SAMNAR são usados como base ao SG-PKM. O módulo de comunicação, por exemplo, é representado pelos protocolos de comunicação entre nós e processos internos. Já o módulo de coleta é utilizado para receber as informações do ambiente, filtra-las e encaminha-las para os componentes apropriados.

No SG-PKM, cada nó em um grupo possui um nível de prevenção que representa a probabilidade dele ser comprometido. Esse nível de prevenção é usado pelos demais nós no momento da formação dos grupos e na emissão de certificados entre os grupos. Essa medida é importante pois, se qualquer nó em um grupo está comprometido, ele pode diminuir a resistência, tolerância e eficácia do sistema de gerenciamento de chaves. No SG-PKM, o nível de prevenção de um dado grupo IG_α é definido como o menor valor de prevenção entre todos os membros desse grupo. A Figura 3.4 ilustra um grafo de certificados de grupos, no qual os pesos dos vértices representam os níveis de prevenção dos seus respectivos grupos.

Da mesma forma, as arestas em um grafo de certificados de grupos, Figura 3.4, também possuem pesos. Esses pesos representam a reputação mínima que os membros do grupo emissor possuem no certificado que está sendo emitido. Nesse caso, cada nó do grupo

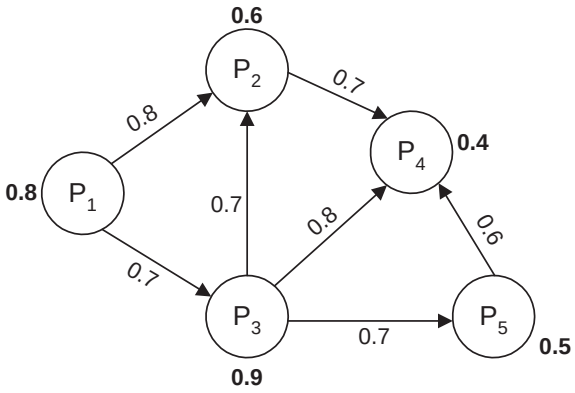


Figura 3.4: Grafo de certificados de grupos

emissor informa o menor valor de reputação que ele possui com relação ao membros do outro grupo. O menor valor de reputação encontrado é definido como o valor da aresta, ou nível de reputação do certificado emitido, e é representado no grafo como o peso da aresta.

Esses valores de prevenção e reputação são informados ao componente de controle da arquitetura SAMNAR, que os utiliza para a tomada de decisões e para a definição dos parâmetros do componente de adaptação. Um desses parâmetros pode ser utilizado na autenticação de chave, por exemplo: um dado nó x_i autentica um dado certificado de grupo $C_{SK_\alpha}^{IG_\beta}$ somente se encontrar uma cadeia de certificados com todos os pesos das arestas maior que $0,8$. Esses valores também podem ser utilizados, por exemplo, na troca dos certificados, em que um dados nó x_i realiza a troca certificados apenas com nós que possuem um nível de prevenção maior que $0,7$. Esses valores de limiar são autoajustáveis e definidos de acordo com as características do ambiente.

Embora este trabalho tenha considerado o uso da arquitetura SAMNAR como suporte à implementação do SG-PKM, qualquer outra arquitetura pode ser utilizada, desde que ofereça serviços semelhantes, como os níveis de prevenção e reputação.

3.12 Conclusão

Neste capítulo, foi apresentada a proposta de um novo esquema de gerenciamento de chaves para MANETs que seja tolerante aos ataques de falta de cooperação e mais

resistente que o PGP-*Like* aos ataques *Sybil*. Esse esquema, chamado de SG-PKM, é totalmente distribuído e auto-organizado, não sendo necessária a presença de uma autoridade central nem mesmo antes da formação da rede. No SG-PKM, os nós são organizados em grupos e, nesses grupos, os nós emitem certificados mutuamente. Além disso, um grupo pode emitir certificados para outros grupos.

Como apresentado, a autenticação de certificados no SG-PKM implementa um mecanismo de resistência a ataques. Sempre que um nó deseja autenticar um certificado de outro nó que não pertence aos seus grupos, ele precisa criar no mínimo duas cadeias disjuntas de certificados, que conectem os seus grupos ao grupo emissor do certificado. Dessa forma, o SG-PKM adiciona um nível de proteção à rede, que dificulta a ação de nós maliciosos.

Além disso da autenticação dos certificados, foram discutidas e detalhadas outras operações de gerenciamento de chaves do SG-PKM, como a entrada de nós na rede, criação dos grupos, atualização e revogação dos certificados. Por fim, foi apresentado o funcionando da arquitetura SAMNAR e como ela foi aplicada no SG-PKM. No próximo capítulo será apresentada uma análise do funcionamento do SG-PKM e o impacto dos ataques de falta de cooperação e *Sybil* em suas operações.

CAPÍTULO 4

AVALIAÇÃO DO GERENCIAMENTO DE CHAVES PÚBLICAS SOBREVIVENTE BASEADO EM GRUPOS

Neste capítulo é apresentada uma avaliação do Gerenciamento de Chaves Públicas Sobrevivente baseado em Grupos para MANETs (*Survivable Group-based Public Key Management for MANETs* (SG-PKM)). Inicialmente, a Seção 4.1 avalia a praticabilidade das suposições feitas pelo SG-PKM quanto à formação dos grupos, considerando as características das redes sociais. A Seção 4.2 apresenta a sobrecarga de comunicação imposta nas operações do esquema. Por fim, a Seção 4.3 mostra, por meio de simulações, a sobrevivência do SG-PKM diante de ataques de falta de cooperação e *Sybil*, quantificando a sua eficácia diante desses ataques.

4.1 Análise da formação dos grupos

Os relacionamentos de confiança assumidos pelo SG-PKM, baseados nas relações de amizades entre os usuários, fornecem o suporte para muitas suposições e operações do esquema. Esses relacionamentos de confiança são a base para a formação dos grupos e para a existência das relações redundantes entre esses grupos. Embora a formação dos grupos seja um requisito para um nó participar da Infraestrutura de Chaves Públicas (ICP), a possibilidade da existência de tais grupos em uma rede social de amigos é analisada nesta seção.

Em todas as análises foi utilizado um exemplo prático de rede social de amigos, o PGP. Como no modelo de confiança assumido pelo SG-PKM, no PGP as chaves públicas são trocadas de uma forma auto-organizada e os certificados são emitidos baseado em um relacionamento de amizade dos usuários. Čapkun, Buttyán e Hubaux (2002) [8] demonstraram que a rede formada pelas chaves públicas e certificados do PGP reflete as relações sociais existentes entre os usuários, e apresenta os fenômenos *small world* e *scale*

free.

Para analisar a viabilidade da existência dos grupos e das relações redundantes entre eles, foi utilizada uma base de dados do PGP (disponível em http://keyring.debian.org/), e foi aplicada a metodologia e as métricas propostas por Latapy, Magnien e del Vechio [39]. Inicialmente, a base de dados do PGP foi analisada como um grafo simétrico $G_{sim} = (V, E)$, em que V é o conjunto de chaves de públicas representando os vértices, e E é o conjunto de certificados representando as arestas. Em seguida, foram extraídos os cliques máximos de G_{sim}. Cliques em um grafo representam um subconjunto de vértices, de forma que quaisquer dois vértices são conectados por uma aresta [28]. Um clique é chamado máximo se ele não está dentro de nenhum outro clique [6]. No SG-PKM, os cliques representam os grupos e mostram que todos os nós trocaram simetricamente as suas chaves públicas.

A Tabela 4.1 apresenta as estatísticas sobre os cliques no grafo do PGP. Na análise desse grafo foi utilizado o algoritmo proposto em [61] para encontrar os cliques. Em seguida, foram comparados os números dos cliques gerais e cliques máximos. Em geral, o número de cliques com um tamanho igual a 3, 4, 5 ou 6 é maior do que os outros. Esses resultados confirmam a possibilidade das formações de grupos usando como base os gráficos do PGP. Dessa forma, a primeira suposição assumida pelo SG-PKM, relacionada à formação de grupos baseados nas relações de amizade dos usuário, é possível de ser alcançada nessas redes.

Tabela 4.1: Estátisticas dos cliques para o grafo do PGP com $|V| = 956$ e $|E| = 14647$

Tamanho do clique	**# de cliques**	**# de cliques máximos**
1	956	9
2	14647	1921
3	47661	4460
4	78016	6599
5	77160	6395
6	49150	4893
9	716	351

4.2 Custo de comunicação

Nesta seção, a sobrecarga de comunicação gerada pelo SG-PKM nas operações de autenticação, revogação e atualização de certificados é analisada. Todos esses custos de comunicação são medidos considerando a quantidade de mensagens trocadas entre os nós.

4.2.1 Autenticação

No SG-PKM, quando o nó x_i deseja autenticar um certificado $C_{SK_\beta}^{x_v}$, a maioria das operações são realizadas localmente, por x_i. Como apresentado na Seção 3.7, inicialmente o nó x_i procura por duas cadeias de certificados válidas em G_i, ligando os seus grupos ao grupo IG_β. Essa operação não gera um custo de comunicação adicional ao sistema. Caso $\nexists(PK_\alpha \rightrightarrows PK_\beta) \in G_i : x_i \in IG_\alpha$, então ele precisa solicitar os certificados atualizados do nó x_v e criar $G_1 = G_i \cup G_v$. Caso $\exists(PK_\alpha \rightrightarrows PK_\beta) \in G_1 : x_i \in IG_\alpha$, então a sobrecarga de comunicação para a autenticar o certificado $C_{SK_\beta}^{x_v}$, denotada por $ACO(C_{SK_\beta}^{x_v})$ - *Authentication Communication Overhead*, é:

$$ACO(C_{SK_\beta}^{x_v}) = (UR\_Req + m.UR\_Rep) \; . \; \Delta h_{x_i - x_v} \tag{4.1}$$

em que $\Delta h_{x_i - x_v}$ é a média do número de saltos entre os nós x_i e x_v, e UR_Req e UR_Rep representam, respectivamente, as mensagens de pedido e resposta dos certificados do repositório atualizado do nó x_v.

Porém, caso $\nexists(PK_\alpha \rightrightarrows PK_\beta) \in G_1 : x_i \in IG_\alpha$, o nó x_i usará as informações do seu repositório de certificados não-atualizados, formando $G_2 = G_i \cup G_i^N$. Se $\exists(PK_\alpha \rightrightarrows PK_\beta) \in G_2 : x_i \in IG_\alpha$, então, para cada certificado de grupo não-atualizado usado para formar as duas cadeias, o nó x_i deve solicitar a validação para os membros do grupo emissor desse certificado. Assim, o custo total de autenticação depende da quantidade de certificados de grupos não-atualizados presentes nas cadeias encontradas. No SG-PKM, a sobrecarga de comunicação para validar um dado certificado de grupo $C_{SK_\delta}^{IG_\omega}$, em número de mensagens, denotado por $VCO(C_{SK_\delta}^{IG_\omega})$ - *Validation Communication Overhead*, é:

$$VCO(C_{SK_\delta}^{IG_\omega}) = (m.VREQ + m.VREP)\ .\ \Delta h \tag{4.2}$$

em que Δh é a média do número de saltos entre os nós, e VREQ e VREP representam, respectivamente, as mensagens de pedido e resposta de validação de um certificado de grupo.

Por fim, a sobrecarga total para autenticar um certificado $C_{SK_\beta}^{x_v}$, denotada por $TACO(C_{SK_\beta}^{x_v})$ - *Total Authentication Communication Overhead*, no pior caso, é:

$$TACO(C_{SK_\beta}^{x_v}) = ACO(C_{SK_\beta}^{x_v}) + k.VCO(C_{SK_\delta}^{IG_\omega}) \tag{4.3}$$

em que k é a quantidade de certificados não-atualizados encontrados nas cadeias de certificados de grupos, necessários para autenticar a chave pública do grupo IG_β.

4.2.2 Revogação

Como apresentado na Seçao 3.10.1, caso um dado nó x_i deseje revogar um certificado de um dado nó x_j, ambos membros de grupo IG_α, o nó x_i deve enviar uma mensagem de pedido de revogação de certificado $C_{SK_\alpha}^{x_j}$ para todos os demais membros do grupo IG_α. Ele aguarda, então, por no mínimo t mensagens de aceitação da revogação do certificado e, em seguida, envia uma mensagem informando da revogação do certificado para todos os membros do grupo IG_α e para todos os membros dos grupos (IG_β) que emitiram um certificado para IG_α.

Os membros dos grupos que emitiram um certificado para o grupo IG_α propagam essa mensagem para todos os nós que solicitaram uma validação do certificado do grupo IG_α, informando da presença de um certificado revogado nesse grupo. Sendo L a lista dos nós que solicitaram uma validação do certificado do IG_α, a sobrecarga de comunicação para o nó x_i revogar o certificado $C_{SK_\alpha}^{j}$, em número de mensagens, denotada por $RCO(C_{SK_\alpha}^{j})$ - *Revocation Communication Overhead*, é:

$$RCO(C_{SK_\alpha}^{x_j}) = (3(|IG_\alpha - x_i|) + |IG_\beta : IG_\beta \rightarrow IG_\alpha \in G| + |L|)\ .\ \Delta h \tag{4.4}$$

em que a constante 3 representa as três mensagens trocadas entre x_i e os demais membros do grupo IG_α.

O custo total depende da quantidade de nós que solicitaram uma validação do certificado que está sendo revogado.

Para revogar explicitamente um certificado de grupo, um membro do grupo emissor do certificado solicita a todos os demais membros desse grupo, a revogação do certificado. Então, ele aguarda pela resposta de pelo menos t nós aceitando a revogação do certificado. Em seguida, esse nó envia uma mensagem para todos os outros membros do seu grupo e para todos os demais nós que solicitaram uma validação desse certificado. Sendo L a lista dos nós que solicitaram uma validação do certificado $C_{SK_\alpha}^{IG_\beta}$ que está sendo revogado, a sobrecarga de comunicação para revogar esse certificado, em número de mensagens, denotada por $RCO(C_{SK_\alpha}^{IG_\beta})$, é:

$$RCO(C_{SK_\alpha}^{IG_\beta}) = (3(|IG_\alpha - x_i|) + |L|) \; . \; \Delta h \tag{4.5}$$

Da mesma forma que na revogação dos certificados de nós, o custo total depende da quantidade de nós que solicitaram uma validação do certificado que está sendo revogado.

4.2.3 Atualização

Quando um dado nó x_i deseja obter uma atualização do seu certificado $C_{SK_\alpha}^{x_i}$, ele envia uma mensagem para todos os demais membros do grupo IG_α e aguarda pelo menos t respostas de atualização do seu certificado. Em seguida, ele envia o certificado atualizado para todos os membros de grupo IG_α. Dessa forma, a sobrecarga de comunicação para a atualização de um certificado de nó $C_{SK_\alpha}^{x_i}$, em número de mensagens, denotada por $UCO(C_{SK_\alpha}^{x_i})$ - *Update Communication Overhead*, é:

$$UCO(C_{SK_\alpha}^{x_i}) = (3|IG_\alpha - x_i|) \; . \; \Delta h \tag{4.6}$$

Já para a atualização do certificado do grupo IG_β ($C_{SK_\alpha}^{IG_\beta}$), o nó x_i envia um pedido de atualização do certificado para todos os demais membros do grupo IG_α e aguarda

pelo menos t respostas de atualização do certificado. Em seguida, ele envia o certificado atualizado para todos os membros do grupo IG_α, para todos os membros do grupo IG_β e para todos os nós que solicitaram uma validação do certificado que está sendo atualizado. Sendo L a lista dos nós que solicitaram uma lista dos nós que solicitaram uma validação do certificado $C_{SK_\alpha}^{IG_\beta}$, então a sobrecarga de comunicação para atualizar esse certificado, em número de mensagens, denotada por $UCO(C_{SK_\alpha}^{IG_\beta})$, é:

$$UCO(C_{SK_\alpha}^{IG_\beta}) = (3|IG_\alpha - x_i| + |IG_\beta - IG_\alpha| + |L|) \; . \; \Delta h \tag{4.7}$$

Nesse caso, o custo de comunicação é diretamente proporcional ao número de nós que solicitaram a validação do certificado que está sendo atualizado. Para diminuir ainda mais esse custo, o nó x_i pode enviar essa mensagem apenas para os nós que solicitaram recentemente uma validação desse certificado, ou simplesmente não enviar a atualização, e deixar que os próprios nós verifiquem a atualização desse certificado, quando necessário.

4.3 Simulações

Nesta seção, as métricas e os cenários usados nas simulações para avaliar o desempenho e a eficácia do SG-PKM são apresentados. Nessas simulações, o SG-PKM é submetido a ataques de falta de cooperação e *Sybil*. Em seguida, são discutidos os resultados das simulações.

4.3.1 Métricas

Considerando dois certificados quaisquer, PK_α e PK_β, em um grafo (G) de certificado de grupos, $(PK_\alpha \rightrightarrows PK_\beta)$ representa um cadeia de certificados entre os dois vértices. Já uma associação $(x_i \rightsquigarrow x_j)$ entre dois dados nós, x_i e x_j, representa que o nó x_i é capaz de validar o certificado x_j, ou seja, que o nó x_i é capaz de encontrar no mínimo dois caminhos disjuntos conectando os seus grupos ao grupo emissor do certificado do nó x_j. Na descrição das métricas, também foi considerado V como o conjunto de nós participantes da Infraestrutura de Chaves Públicas (ICP) e IG como o conjuntos dos grupos.

Para avaliar o SG-PKM, foram usadas as seguintes métricas: Troca de Certificados de Grupos (*Certificate Exchange Convergence* (CE)), Taxa de Autenticação de Usuários (*User Authenticability* (UA)), Alcançabilidade dos Grupos (*Group Reachability* (GR)), Grupos Não Comprometidos (*Non-Compromised Groups* (NCG)) e Autenticações Não Comprometidas (*Non-Compromised Authentication* (NCA)). As métricas CE, UA e GR são usadas para avaliar o SG-PKM em cenários com ataques de falta de cooperação, enquanto as métricas NCG e NCA são usadas para avaliar o esquema diante de ataques *Sybil*. Essas métricas são definidas como:

- CE é a percentagem média dos certificados de grupos nos repositórios locais dos nós no tempo t. Ela também representa o tempo necessário para todos os nós tenham todos os certificados de grupos emitidos em seus repositórios. O valor ideal para essa métrica é 100%, contudo algumas condições como a fase de inicialização da ICP, a formação dos grupos, os ataques, entre outras, podem reduzir esse percentagem. CE é definida como:

$$CE(t) = \frac{\sum_{i \in X} CE_i(t)}{|X|} \quad \text{em que}$$

$$CE_i(t) = \frac{\sum_{IG_\alpha, IG_\beta \in IG} (PK_\alpha \rightarrow PK_\beta) \in (G_i \ \cup \ G_i^N)}{\sum_{IG_\gamma, IG_\delta \in IG} (PK_\gamma \rightarrow PK_\delta) \in G} \tag{4.8}$$

- UA é a percentagem média de autenticações de usuário após o tempo de convergência do sistema de gerência de chaves. Essa métrica é quantificada pelas cadeias de certificados encontradas nos repositórios atualizados e não-atualizados de um nó x_i. As autenticações dos usuários são contabilizadas apenas se duas ou mais cadeias de certificados disjuntas são encontradas para autenticar o nó. Sob ataques, essa métrica também indica a sobrevivência do SG-PKM, avaliando se os nós são capazes de autenticarem-se mutuamente mesmo diante de ataques de falta de cooperação.

A métrica UA pode ser definida como:

$$UA = \frac{\sum_{i \in X} UA_i}{|X|} \quad \text{em que}$$

$$UA_i = \sum_{j \in X} (x_i \rightsquigarrow x_j) \in (G_i \cup G_j \cup \ G_i^N) \tag{4.9}$$

- GR é a percentagem média das cadeias de certificados para alcançar os certificados de grupos nos repositórios atualizados e não-atualizados de um dado nó x_i no tempo t. A diferença com relação à métrica UA é que, nesse caso, são quantificados apenas os certificados de grupos, sem a necessidade de encontrar duas ou mais cadeias de certificados disjuntas para a autenticação. Sendo IG_{x_i} os grupos que o nó x_i participa, então GR pode ser definida como:

$$GR(t) = \frac{\sum_{i \in X} GR_i(t)}{|X|} \quad \text{em que}$$

$$GR_i(t) = \sum_{\substack{IG_\alpha \in IG_{x_i} \\ IG_\beta \in IG}} (PK_\alpha \rightsquigarrow PK_\beta) \in (G_i \cup \ G_i^N) \tag{4.10}$$

- NCG é a percentagem de grupos não comprometidos mesmo na presença de nós desonestos na rede. Essa métrica representa a sobrevivência do SG-PKM diante de ataques *Sybil*. Sendo IG o conjunto dos grupos existentes na ICP, a métrica NCG pode ser definida como:

$$NCG = \frac{\sum_{IG_\alpha \in IG} NCG_\alpha}{|IG|} \quad \text{em que}$$

$$NCG_\alpha = \begin{cases} 1 & \text{se} \quad \nexists\ f \in IG_\alpha : f \text{ é uma identidade falsa} \\ 0 & \text{caso contrário} \end{cases} \tag{4.11}$$

- NCA é a percentagem de grupos que não têm suas autenticações de chaves públicas comprometidas por nós desonestos. Essa métrica representa a sobrevivência do

processo de autenticação do SG-PKM diante de ataques *Sybil*. Sendo F o conjunto de nós *Sybil*, NCA pode ser definida como:

$$NCA = \frac{\sum_{i \in X} NCA_i}{|X|} \qquad \text{em que}$$

$$NCA_i = \begin{cases} 1 & \text{se} \quad \nexists \, (pk_i \rightsquigarrow pk_f) \quad \forall f \in F \\ 0 & \text{caso contrário} \end{cases} \tag{4.12}$$

4.3.2 Cenários

Para avaliar o desempenho e a sobrevivência do SG-PKM, foi utilizado o Network Simulator versão 2.30 [48]. As simulações foram realizadas na presença de ataques de falta de cooperação e *Sybil*. Da mesma forma que na avaliação do PGP-*Like* [20], nessas simulações, em um ataque de falta de cooperação, o nó malicioso não colabora com os serviços da ICP, principalmente no mecanismo de troca dos certificados.

Nas simulações, 100 nós usam o IEEE 802.11 com função de coordenação distribuída (*Distributed Coordination Function* (DFC)) como procolo de acesso ao meio. O modelo de propagação é o reflexão no solo em dois raios (*two-ray ground reflection*) e o raio de alcance das antenas é de 50m e 120m. Os nós se movimentam em uma área de 1000m x 1000m e de 1500m x 300m, seguindo o modelo de movimentação aleatória *waypoint*, com velocidades máxima de 5 m/s, 10 m/s e 20 m/s e com tempo de pausa máximo de 20 segundos. O tempo total de simulação é 3000 segundos e os resultados são médias de 35 simulações com 95% de intervalo de confiança.

As chaves públicas e privadas são criadas pelos nós apenas durante a formação dos grupos. Os certificados também são emitidos durante a formação dos grupos e não existe nenhum mecanismo de detecção de nós maliciosos na rede. Sem perder a generalidade, assume-se que todos os nós possuem o mesmo T_{troca} e que os nós não realizam as trocas de certificados de forma síncrona. Dessa forma, se o nó x_i está enviando o seus certificados ao nó x_u, isso não implica que o nó x_u também esteja enviando os seus certificados ao nó

x_i. O intervalo das trocas de certificados é de 60 segundos.

De acordo com a Tabela 4.1, as redes sociais apresentam um grande número de cliques com um tamanho igual a 3, 4, 5 e 6. Dessa forma, o SG-PKM foi avaliado variando o tamanho dos grupos (m) entre 3 e 6. Foram criados, em média, 496 grupos com 3 membros, 253 grupos com 4 membros, 63 grupos com 5 membros e, por fim, 17 grupos com 6 membros. Note que devido à dificuldade de formar grupos com 6 membros, é possível que alguns nós não consigam participar de nenhum grupo ou que alguns grupos fiquem isolados. Porém esses nós não serão considerados na avaliação do SG-PKM. O objetivo é verificar o impacto dos tamanho dos grupos na sua eficácia e sobrevivência. Por simplicidade, as relação de confiança dos usuários são formadas como em [63].

A Tabela 4.2 apresenta uma comparação entre os valores relevantes para estes testes encontrados nos grafos do PGP e os grafos gerados. Como na Seção 4.1, foi aplicada a metodologia e as métricas propostas por Latapy, Magnien e del Vechio [39] sobre uma base dados do PGP e os grafos gerados neste trabalho. Foram considerados os seguintes parâmetros:

a. o coeficiente de aglomeração (*clustering*), que é a probabilidade dos vértices de um grafo formarem um clique;

b. a redundância entre os cliques, que é a fração de pares vizinhos de um grupo IG_α ligados a outro grupo diferente de IG_α;

c. a distância entre os nós, que é o tamanho média das cadeias de relacionamento entre dois nós quaisquer em um grafo.

Tabela 4.2: Comparação dos parâmetros entre os grafos do PGP e os grafos gerados

Parâmetros	**Grafos PGP**	**Grafos gerados**
coeficiente de aglomeração	0.030	0.037
redundância entre os cliques	0.213	0.282
distância entre os nós	3.739	3.726

É possível notar que os valores são similares nos grafos do PGP e nos grafos gerados

para a avaliação do SG-PKM. Isso mostra que os grafos usados apresentam o comportamento social esperado.

4.3.3 Ataques de falta de cooperação

Inicialmente, foi comparada a eficácia do SG-PKM com o PGP-*Like* diante de ataques de falta de cooperação considerando a métrica CE. Como na avaliação do PGP-*Like* foram considerados cenários sem ataques e com 5%, 10%, 20% e 40% de nós egoístas. Esses nós egoístas emitem certificados e formam grupos, mas não cooperam no mecanismo de troca de certificados, i.e. eles solicitam e armazenam certificados em seus repositórios locais, mas não respondem aos pedidos para trocas de certificados dos demais nós.

Nesta seção são apresentados os resultados das simulações realizadas em cenários de 1000 x 1000 metros. Devido à similaridade de comportamento, os resultados das simulações realizadas com cenários 1500 x 300 metros encontram-se no Apêndice B. Como no caso do PGP-*Like* a mudança no tamanho do ambiente afeta apenas o tempo necessário para a convergência das trocas de certificados.

A Figura 4.1 mostra uma comparação do PGP-*Like* e o SG-PKM com grupos de tamanho 3, 4, 5 e 6, em um cenário com velocidade máxima de 5 m/s e 120 metros de raio de alcance. No SG-PKM, a métrica CE alcança o seu ponto de convergência antes que o PGP-*Like*, independentemente do número de nós malcomportados. Quando m é igual a 5, CE alcança 100% aproximadamente após 500 segundos do tempo de vida da rede. Já para m igual a 3 e 4, 100% de CE é alcançado antes dos 350 segundos do tempo de vida da rede. Por fim, quando m é igual a 6, CE alcança 100% aproximadamente após 700 segundos de vida da rede.

Enfatizando, quanto maior é o valor de CE, maior é a probabilidade de um nó encontrar um caminho de certificados de grupos em seu repositório na fase de autenticação. Contudo, isso não significa que todos os grupos serão capazes de autenticar todos os demais certificados de grupos, devido à necessidade de redundância na autenticação.

A Figura 4.2 apresenta uma comparação do PGP-*Like* e o SG-PKM em um cenário com velocidade máxima de 10 m/s e 120 metros de raio de alcance. Novamente, o SG-

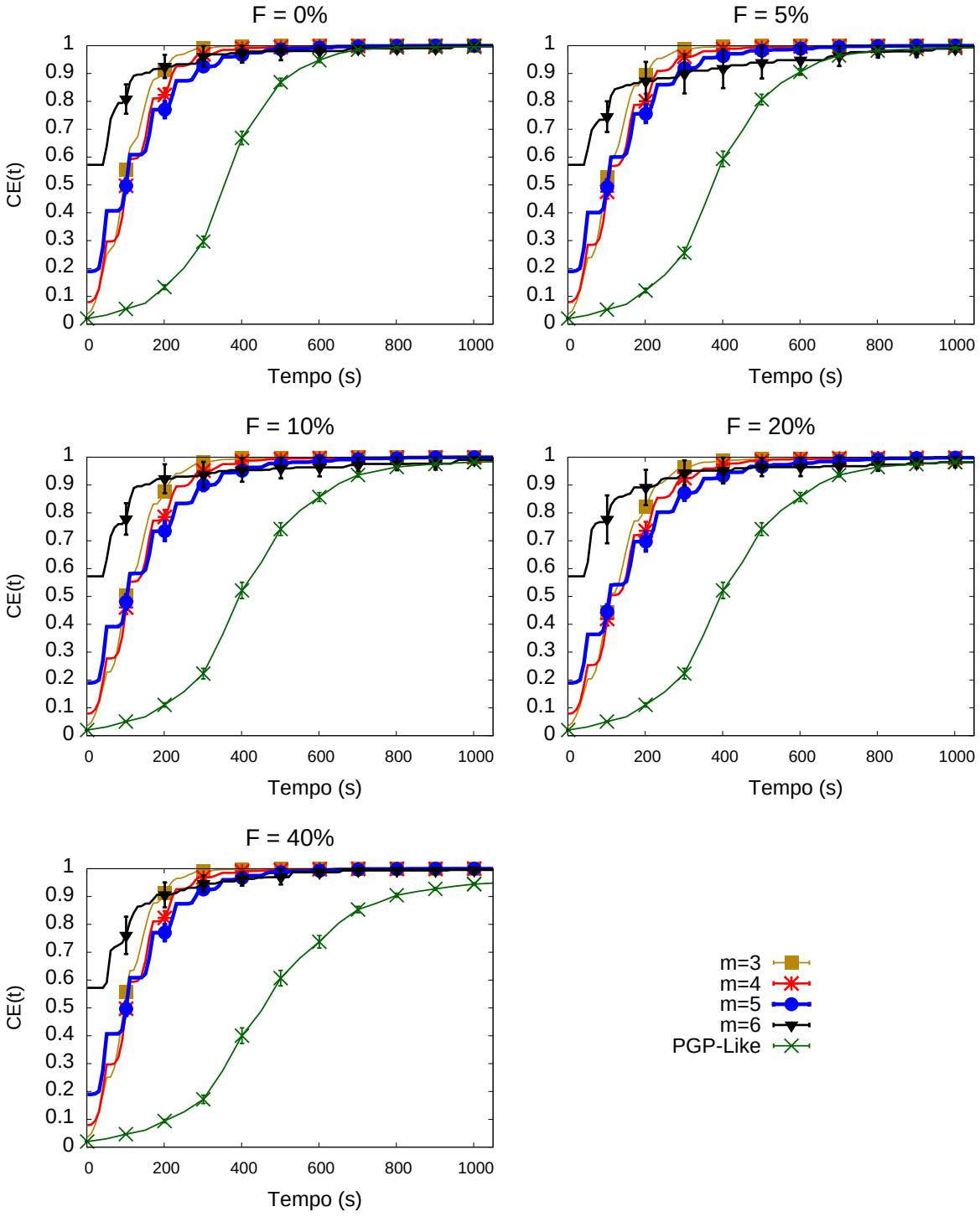


Figura 4.1: Tempo de convergência com velocidade de 5 m/s e raio de 120 metros

PKM precisa de um tempo menor que o PGP-*Like* para a convergência das trocas de certificados. Isso ocorre devido à formação dos grupos, no qual mais nós possuem um mesmo certificado armazenado, aumentando a redundância das informações. Mesmo na presença de 40% de nós egoístas, independente do tamanho dos grupos, CE sempre

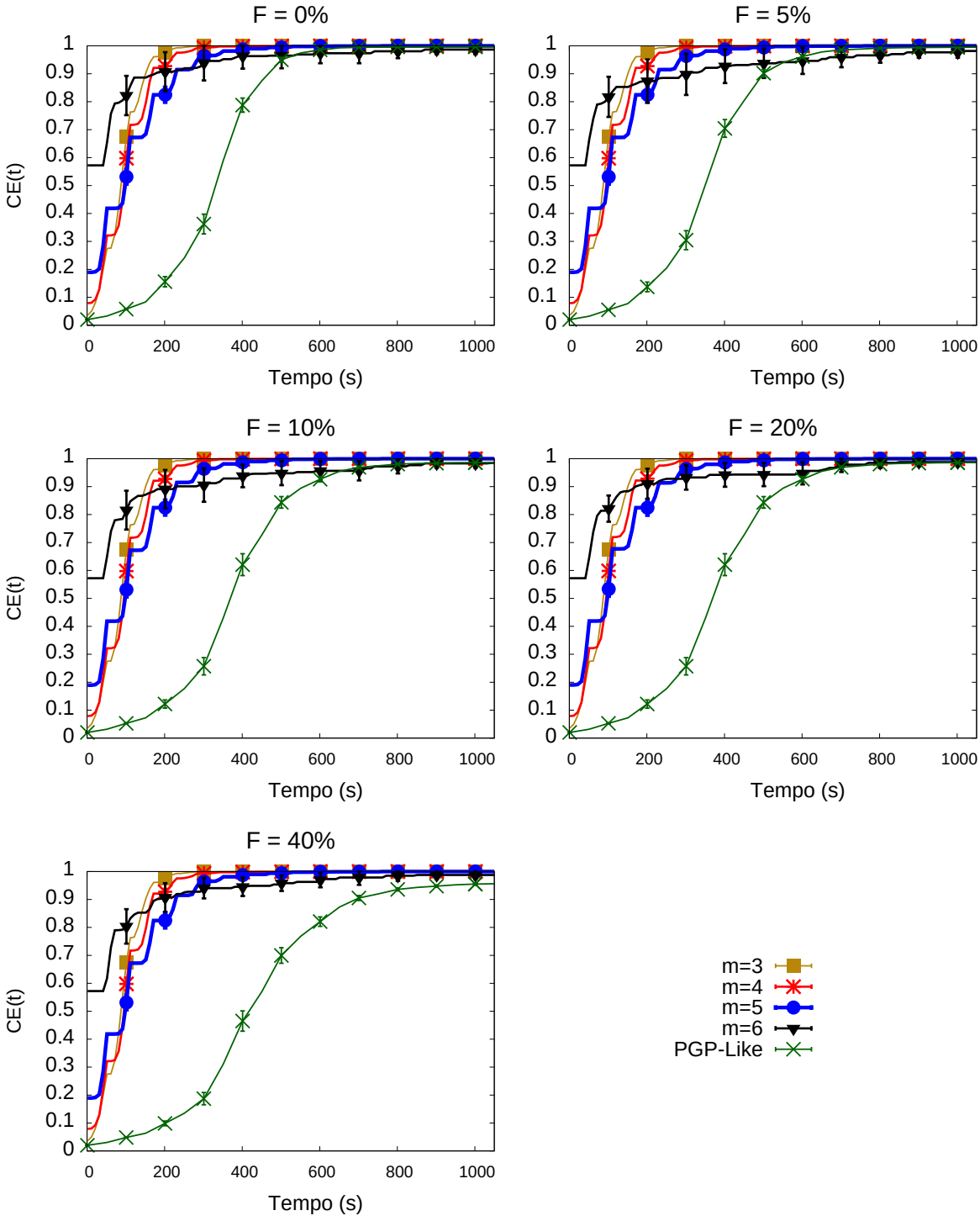


Figura 4.2: Tempo de convergência com velocidade de 10 m/s e raio de 120 metros

alcança 100%. Quando m é igual a 5, por exemplo, CE alcança 100% aproxidamente após 400 segundos de vida da rede. Quando m é igual a 3 ou 4, esse valor é alcançado antes dos 300 segundos de vida da rede.

A Figura 4.3 mostra uma comparação do SG-PKM com o PGP-*Like* em cenários

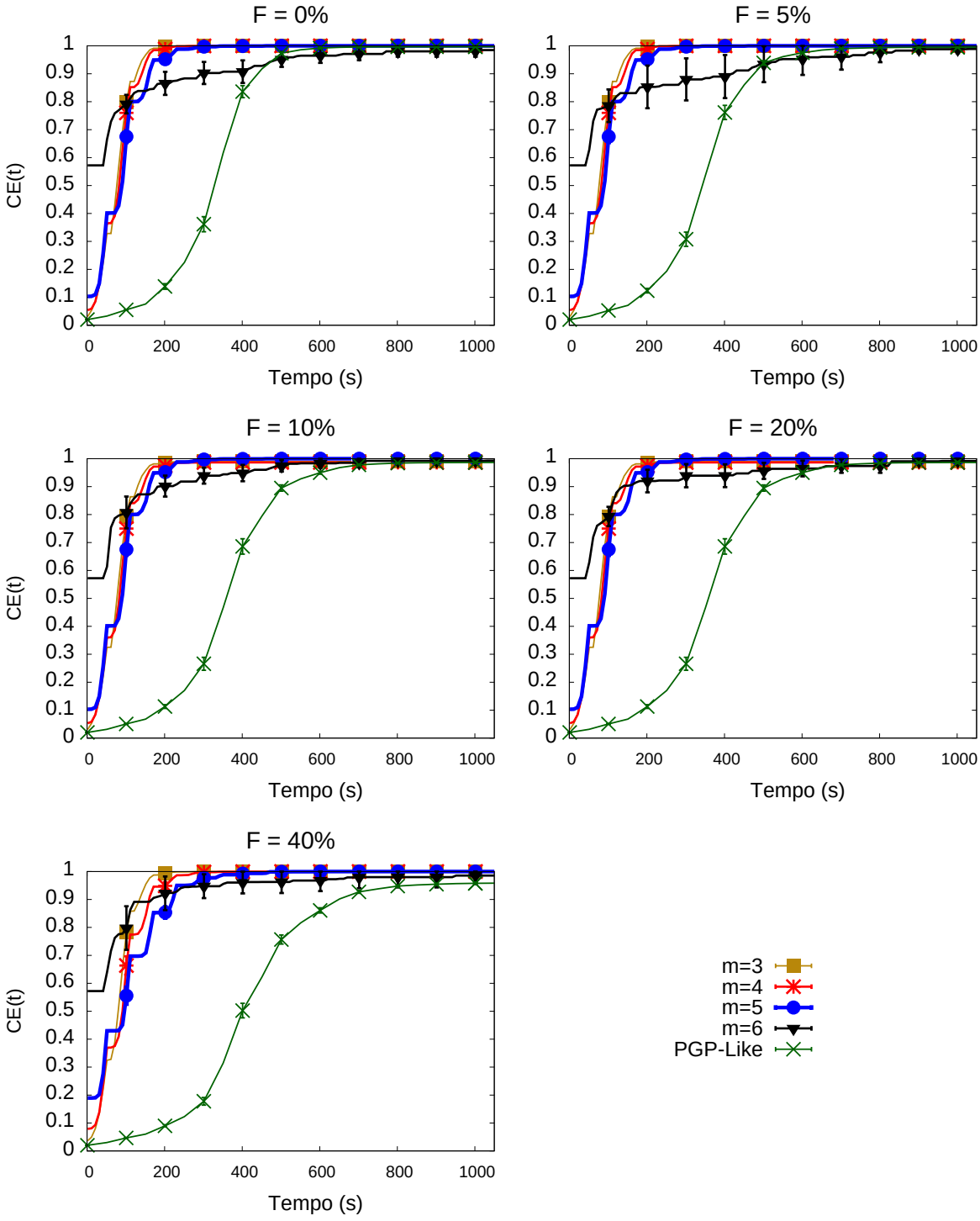


Figura 4.3: Tempo de convergência com velocidade de 20 m/s e raio de 120 metros

velocidade máxima de 20 m/s e 120 metros de raio de alcance. No SG-PKM, como nos casos anteriores, CE sempre alcança 100%. Além disso, em todos os casos, o SG-PKM alcança o seu ponto de convergência antes que o PGP-*Like*, independentemente do tamanho dos grupos e do número de nós egoístas. Em cenário com até 20% de atacantes

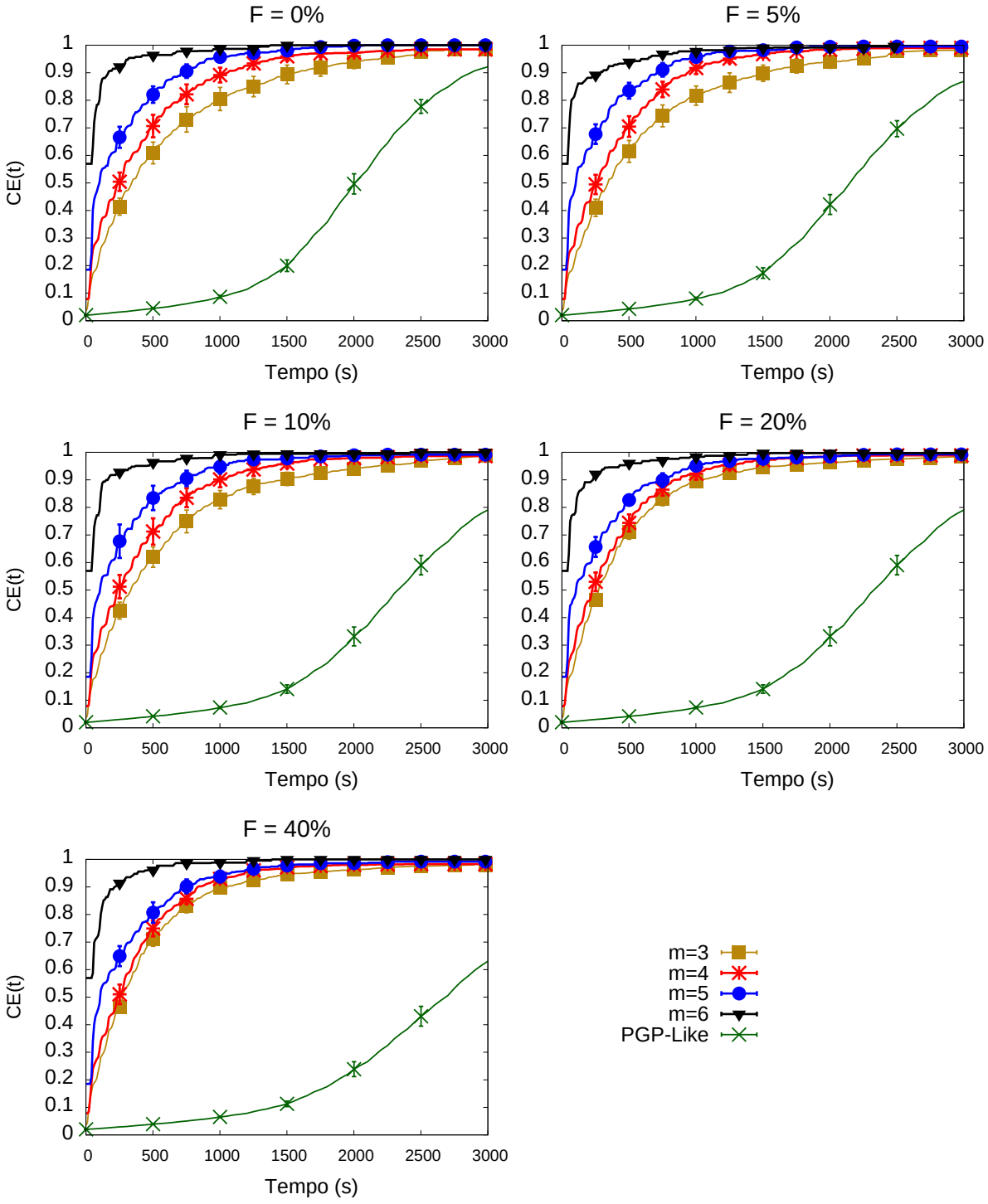


Figura 4.4: Tempo de convergência com velocidade de 5 m/s e raio de 50 metros

e m igual a 3, 4, ou 5, CE alcança 100% antes dos 300 segundos de vida da rede.

A Figura 4.4 mostra uma comparação do PGP-*Like* com o SG-PKM em cenários com velocidade máxima de 5 m/s e 50 metros de raio de alcance. No SG-PKM, o ponto de convergência é alcançado antes que o PGP-*Like*, independentemente do número de nós

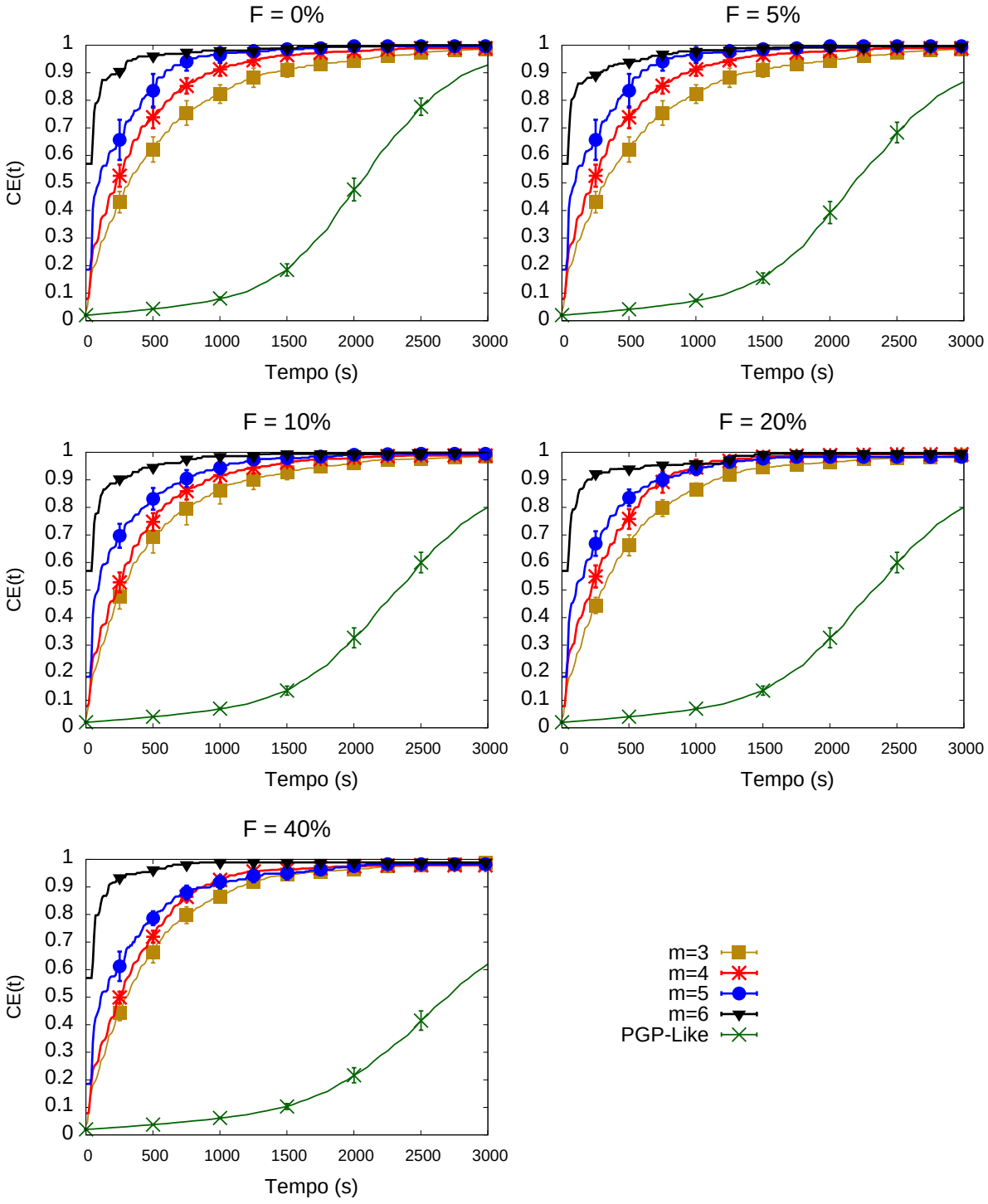


Figura 4.5: Tempo de convergência com velocidade de 10 m/s e raio de 50 metros

malcomportados. Nos cenários com m igual a 4, 5 ou 6, CE alcança 100% antes de 1500 segundos, e com m igual a 3 esse valor é alcançado após 2500 segundos.

Já a Figura 4.5 apresenta essa comparação em um cenário com velocidade máxima de 10 m/s e 50 metros de raio de alcance. Como nos casos anteriores, o SG-PKM precisa

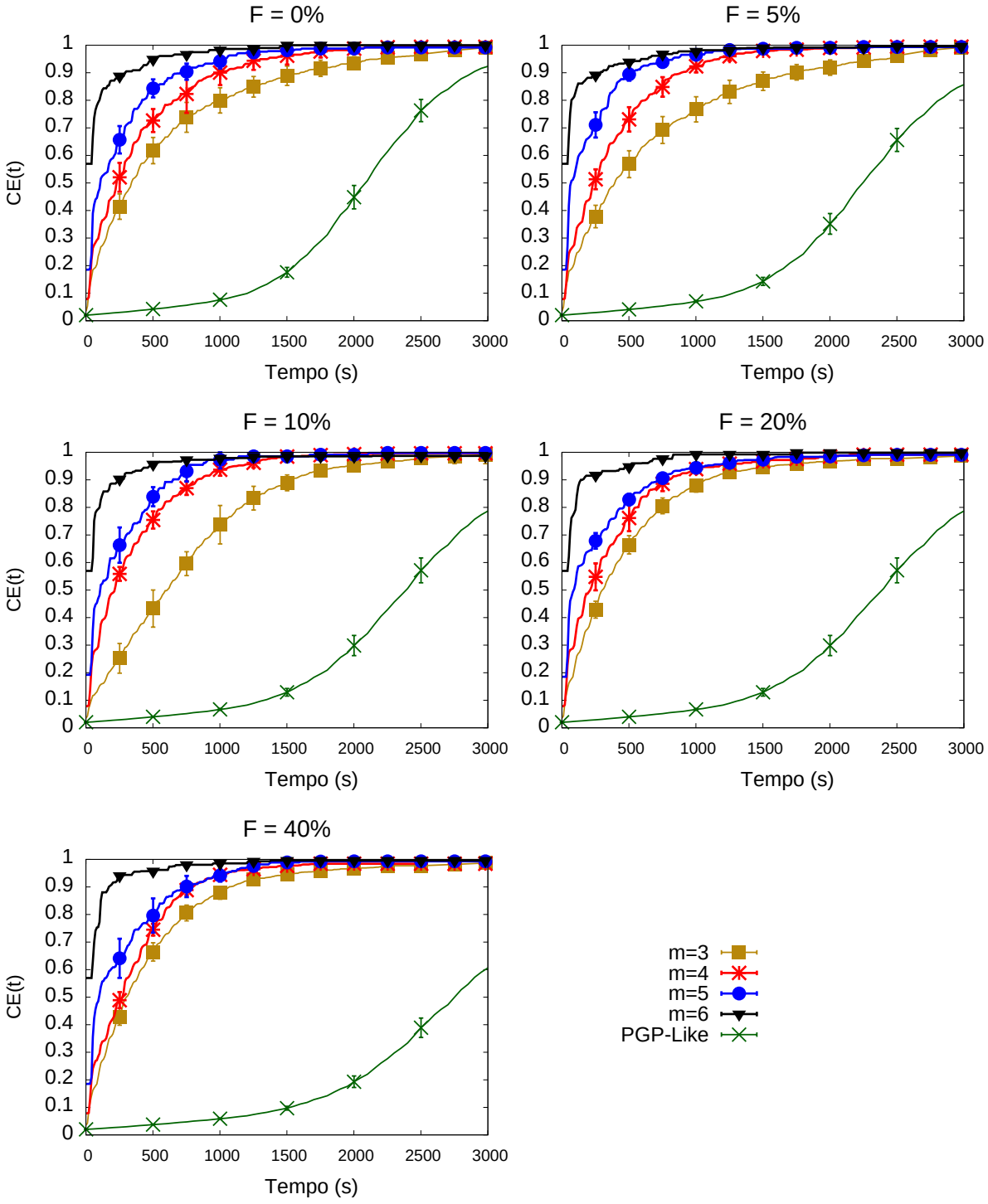


Figura 4.6: Tempo de convergência com velocidade de 20 m/s e raio de 50 metros

de um tempo bem menor que o PGP-*Like* para a convergência das trocas de certificados. Em todos os casos CE alcança 100%.

Por fim, a Figura 4.6 mostra uma comparação do SG-PKM com o PGP-*Like* em cenários com velocidade máxima de 20 m/s e 50 metros de raio de alcance. No SG-PKM,

como nos casos anteriores, CE alcança o seu ponto de convergência antes que o PGP-*Like*, independentemente do tamanho dos grupos e do número de nós egoístas. Em cenário com até 20% de atacantes e m igual a 4, 5 ou 6, CE alcança 100% antes dos 1500 segundos de vida da rede.

Como apresentado, em todos os casos, independente da velocidade de movimentação dos nós, do raio de alcance das antenas e da quantidade de nós malcomportados, o valor de CE para o SG-PKM sempre alcança 100%. O SG-PKM não é afetado pelo ataque de falta de cooperação e apresenta um tempo de convergência das trocas de certificados menor que o PGP-*Like*, devido à formação dos grupos e à emissão de certificados entre os grupos, garantindo uma redundância de armazenamento dos certificados emitidos.

A Figura 4.7 apresenta os resultados para a métrica GR em cenários com 0%, 5%, 10%, 20% e 40% de atacantes, após a convergência das trocas de certificados. Como esses resultados são calculados após a convergência das trocas de certificados, eles são os mesmos para cenários com tamanho 1000 x 1000 metros e 1500 x 300 metros, e para raios de alcance de 50 metros e 120 metros. Como esperado, independente da percentagem de atacantes é observado que GR apresenta o mesmo comportamento. Em todos casos, após a convergência do sistema, GR é praticamente 100%. Mesmo quando esse valor não

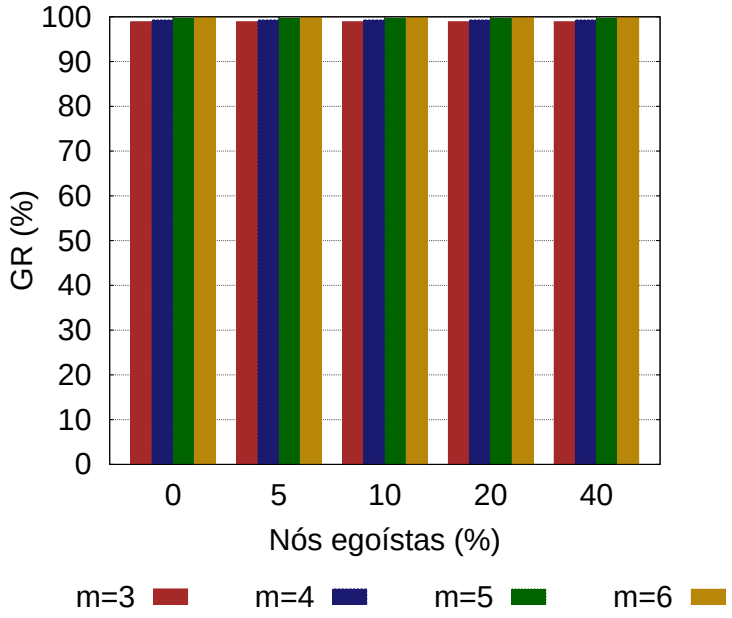


Figura 4.7: Alcançabilidade dos grupos após a convergência das trocas de certificados

alcança 100%, ele fica acima dos 99%. A Figura 4.8 apresenta os mesmos resultados, entre o intervalo de 95% a 100%, para ressaltar os resultados. Note que a quantidade de atacantes não afeta a alcançabilidade dos grupos. Como um mesmo certificado de grupo é armazenado por vários nós (membros de grupo emissor e do grupo associado ao certificado), mesmo diante de vários nós egoistas é possível se obter um caminho de certificados para alcançar a maioria dos grupos do sistema.

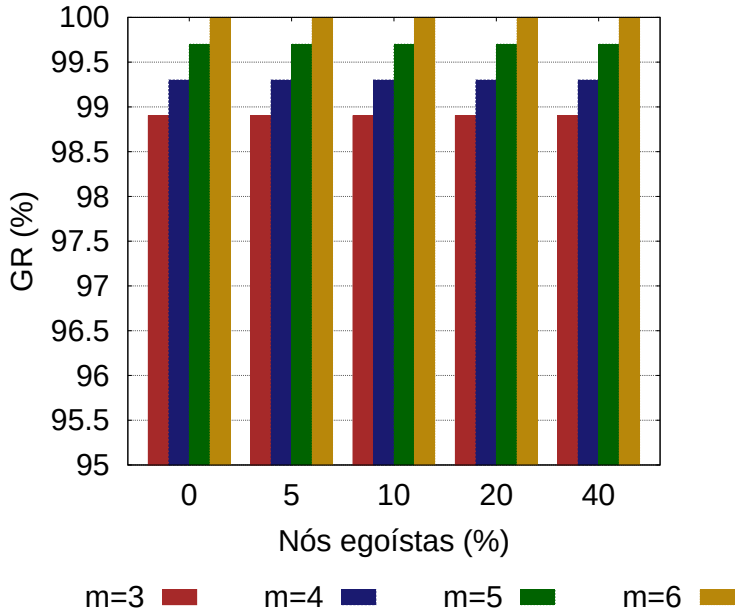


Figura 4.8: Alcançabilidade dos grupos após a convergência das trocas de certificados (visão detalhada)

Embora GR seja 100% em todos os casos, e independente da quantidade de nós egoístas, isso não significa que todos os usuário podem autenticar-se mutuamente. Relembrando, para que um nó possa autenticar o certificado de chave pública de outro nó, que não faz parte de nenhum dos seus grupos, ele precisa formar no mínimo duas cadeias disjuntas de certificados grupos, entre os seus grupos e o grupo emissor do certificado.

A Figura 4.9 compara os valores de UA, após o tempo de convergência do sistema, considerando diferentes tamanhos de grupos e percentagem de nós malcomportados. Os resultados mostram que UA apresenta os mesmos valores, independente da quantidade atacantes. É possível observar, também, a forte influência do tamanho dos grupos na percentagem de autenticações. Os resultados mostram que enquanto o tamanho do grupo aumenta, a percentagem de autenticações de usuários também aumenta. Quando m é igual

a 6, o SG-PKM alcança mais que 85% de autenticações válidas de usuários, enquanto que quando m é igual a 3, esse valor é cerca de 40%. Isso ocorre porque com grupos pequenos é mais difícil criar redundâncias entre os grupos e, consequentemente, formar cadeias de certificados disjuntas entre dois grupos quaisquer.

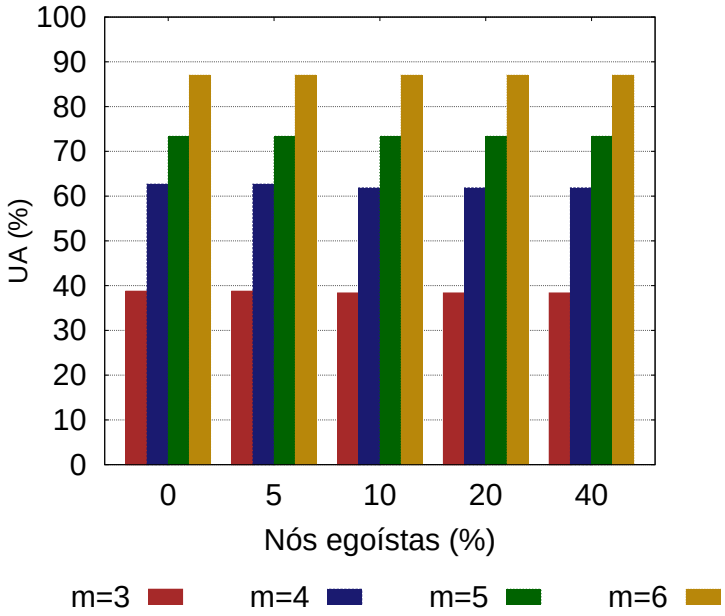


Figura 4.9: Autenticação dos usuário em cenários com ataques de falta de cooperação

Além disso, os resultados também mostram que uma maior percentagem de atacantes não resulta em uma redução de UA quando comparado com resultados sem ataques. Esse comportamento mostra a sobrevivência do SG-PKM aos ataques de falta de cooperação.

4.3.4 Ataques Sybil

A eficácia do SG-PKM também foi avaliada em cenários com a presença de nós *Sybil*. Nesse caso, as métricas NCG e NCA foram consideradas. Como na avaliação do PGP-*Like* foram considerados cenários sem ataques e com 5%, 10%, 20% e 40% de nós maliciosos. Esses nós maliciosos criam identidades falsas ou personificam identidades autênticas, e formam grupos com essas identidades. Em seguida, eles tentam comprometer os nós autênticos, para que esses nós emitam certificados para os grupos falsos.

O objetivo dos nós maliciosos é comprometer um grande número de nós da ICP. Como consequência, se dois nós de um mesmo grupo estão comprometidos, esse grupo pode emi-

tir um certificado para um grupo falso. Quanto maior o número de grupos comprometidos, maior é a probabilidade de uma identidade falsa ser autenticada por um nó válido. O ideal é que o SG-PKM consiga resistir à presença desses nós maliciosos e mantenha os valores de NCG e NCA sempre altos. Relembrando, o PGP-*Like* é totalmente vulnerável a esse tipo de ataque, sempre permitindo que um nó *Sybil* participe das operações do sistema, independente da quantidade de atacantes.

A Figura 4.10 mostra a sobrevivência do SG-PKM aos ataques *Sybil*. Os resultados mostram que com uma percentagem de 5% de atacantes, independentemente do tamanho dos grupos, mais de 90% dos grupos não são afetados. Quando m é igual a 3 esse valor é próximo a 99%. Quando a percentagem de atacantes é 10% e m é igual a 3, NCG é cerca de 95%. Esse valor decresce um pouco quando m é igual a 4 e 5, estando próximo a 90%. Apenas quando m é 6, NCG é pouco menor, porém ainda está próximo a 70%. Isso ocorre porque em grupos maiores, a probabilidade de se encontrar dois ou mais nós maliciosos também é maior e, consequentemente, um nó malicioso é capaz de comprometer mais grupos.

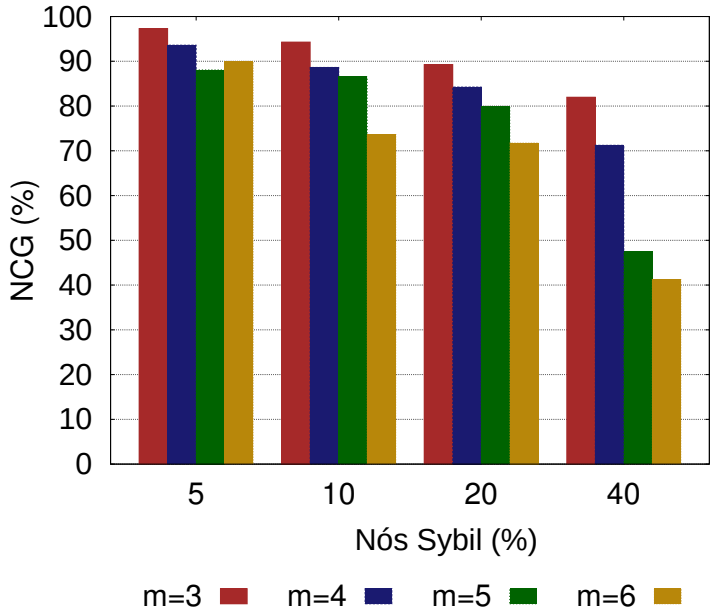


Figura 4.10: Grupos não comprometidos sob ataques Sybil

Já quando a percentagem de atacantes é 20%, mais grupos são levados a emitir certificados a um grupo falso, mas os resultados ainda mostram a sobrevivência do SG-PKM. Quando m é igual a 3, quase 90% dos grupos não estão afetados, e quanto m é igual a 4 e

5, esse valor é cerca de 85% e 80%, respectivamente. Apenas quanto m é igual a 5 ou 6 e o percentual de atacantes é de 40% NCG, apresenta um valor menor, aproximadamente 48% e 41%, respectivamente. Dessa forma, quanto menor o tamanho dos grupos, menor é a possibilidade de um nó *Sybil* convencer outros grupos a emitir um certificado válido para um grupo composto de nós com identidades falsas.

Finalmente, a Figura 4.11 apresenta o impacto dos ataques *Sybil* e dos tamanhos dos grupos no processo de autenticação. Os resultados mostram que quanto m é igual a 6, a percentagem de nós válidos que não autenticam uma identidade falsa é cerca de 98%, quando o sistema está sob 5% de nós atacantes. Esse valor é próximo a 97% com 10% de atacantes e maior do que 95% com de 20% de atacantes.

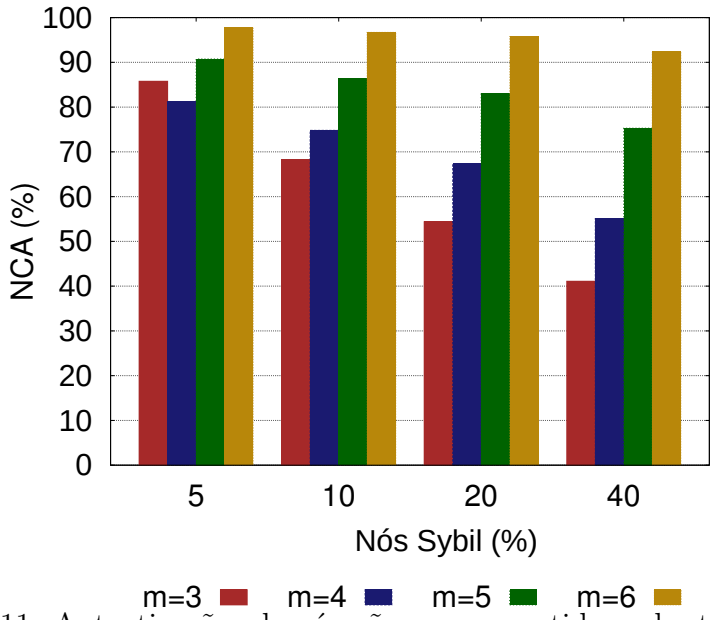


Figura 4.11: Autenticações de nós não comprometidas sob ataques Sybil

Quando o SG-PKM está na presença de 5% de atacantes, a percentagem de nós válidos que não autenticam uma identidade falsa é maior do que 80%. Quando m é igual a 5 esse valor é cerca de 90%. Quando a percentagem de atacantes é 10% e m é 5 ou 6, NCA é, ainda, maior do que 80%. Com m igual a 3 ou 4, esse valor é 74% e 68%, respectivamente. Já quando o sistema está sob um grande número de atacantes (40%), NCA apresenta um valor menor que 70%, para m igual a 3 ou 4. Mas com m igual a 5 esse valor é maior do que 75% e com m igual a 6 ele é cerca de 92%. Nesse caso, quanto maior o tamanho dos grupos, maior é a sobrevivência do SG-PKM. Isso ocorre porque, quanto menores são os

grupos, mais certificados são emitidos entre os grupos, o que facilita a autenticação de um identidade falsa, por membros de outros grupos.

Esses resultados mostram a eficácia do SG-PKM na presença de nós *Sybil*. Relembrando, independente da quantidade de nós atacantes, o PGP-*Like* é totalmente vulnerável à ação maliciosa desses nós. Já o SG-PKM consegue resistir melhor a esse ataque, principalmente com grupos de tamanho 5 ou 6. Nesses casos, uma grande quantidade de nós não-comprometidos não realizam a autenticação de um nó falso (NCA), mesmo que este consiga participar de algum grupo (NCG). Embora seja mais difícil se formar grupos de tamanho 6, esses grupos aumentam a sobrevivência do esquema diante dos ataques *Sybil*.

4.4 Conclusão

Neste capítulo foi apresentada uma avaliação do desempenho e da eficácia do SG-PKM em cenários sem ataques, e em cenários com ataques de falta de cooperação e *Sybil*. Inicialmente, uma análise dos grafos do PGP foi realizada, com o objetivo de mostrar se as restrições do SG-PKM, quanto a formação dos grupos e a emissão mútua de certificados é viável para as redes reais. Essa análise mostrou que as redes sociais podem apresentar um comportamento como o assumido pelo SG-PKM.

Em seguida, a sobrecarga de comunicação resultante das operações do SG-PKM foram apresentados. Como discutido, o esquema adiciona um custo de comunicação baixo à rede, visto que, após a fase de troca dos certificados, a maioria das operações pode ser realizada localmente pelos nós. Além disso, as operações de revogação e atualização dos certificados de nós e grupos, que são as mais custosas, são exporádicas e, portanto, não afeta o desempenho do sistema.

Por fim, a eficácia do SG-PKM diante de ataques de falta de cooperação e *Sybil* foi avaliada, por meio de simulações. Os resultados apresentados mostraram que o sistema é sobrevivente aos ataques citados, mantendo o desempenho e a eficácia mesmo em ambientes hostís. Se comparado com o PGP-*Like*, que é totalmente vulnerável a ataques *Sybil*, o SG-PKM se mostrou bem mais resistente. Nesses casos, em cenários com grupos

de seis membros e até 40% de nós *Sybil*, mais que 90% dos nós não-comprometidos não autenticaram uma identidade falsa.

CAPÍTULO 5

CONCLUSÕES

As MANETs são redes formadas espontaneamente, caracterizadas pela ausência de uma infraestrutura fixa e controle centralizado, construídas por unidades móveis, e possuem uma topologia dinâmica. Essas características tornam essas redes altamente vulneráveis a ataques passivos e ativos. A criptografia tem sido a principal ferramenta para garantir a segurança nas redes, e nas MANETs isso não é diferente. Entretanto, as características dessas redes dificultam a implementação de esquemas de gerenciamento de chaves eficazes e seguros. Diversos esquemas de gerência de chaves foram propostos para MANETs. Entre eles, os esquemas totalmente distribuídos e auto-organizados são os mais indicados para as MANETs, por não necessitarem de nenhuma autoridade centralizada, nem mesmo antes da formação da rede.

O *Sistema de Gerência de Chaves Públicas Auto-Organizado para MANETs*, chamado ao longo deste trabalho de PGP-*Like*, é considerado um dos melhores esquemas totalmente distribuídos e auto-organizados proposto para MANETs. Contudo, suas características o tornam suceptível a ataques maliciosos. Neste trabalho, foi apresentada uma avaliação, realizada por meio de simulações, do PGP-*Like* diante de dois tipos de ataques maliciosos: falta de cooperação e Sybil. Os resultados mostraram que o PGP-*Like*, embora consiga resistir a uma grande quantidade de nós egoístas, é totalmente vulnerável aos ataques Sybil. Mesmo na presença de apenas 5% de nós maliciosos, o sistema é totalmente comprometido pela ação dos atacantes.

Com isso, este trabalho propôs um novo esquema de gerência de chaves para MANETs, chamado de SG-PKM, que tem como objetivo manter o seu desempenho na presença de nós egoístas e Sybil. No SG-PKM, os nós formam grupos baseados nas relações de amizades de seus usuários. Nesses grupos, eles trocam suas chaves públicas e emitem certificados mutuamente. Os membros de um grupo também podem emitir certificados

para outros grupos, também baseados nas relações de amizados dos seus usuários. Nesses pequenos grupos de tamanho m, todos os nós possuem o mesmo papel e não é necessária a presença de um líder, o que torna o sistema atrativo para as MANETs.

Cada grupo possui um par de chaves pública e privada, construídas colaborativamente por todos os membros do grupo. A chave pública é disponibilizada para todos os membros da rede, enquanto que a chave privada é distribuída entre os membro do grupo em um esquema de criptografia de limiar. Caso os nós queiram autenticar-se mutamente, eles devem formar cadeias de certificados conectando os grupos a que pertencem. O SG-PKM prevê ainda um esquema de redundância das cadeias de autenticação, visando aumentar a resistência do sistema contra ataques maliciosos.

O SG-PKM foi avaliado em cenários com ataques de falta de cooperação e *Sybil*. Esses cenários também foram utilizados para avaliar a eficácia do PGP-*Like*. Nos cenários com ataques de falta de cooperação, o SG-PKM conseguiu resistir à não cooperação dos nós egoístas, sem ser afetado. Em todos os casos, e independente da quantidade de nós egoístas, a convergência das trocas de certificados acontece bem antes que o PGP-*Like*. Além disso, a alcançabilidade dos grupos ficou, independente dos tamanho dos grupos, muito próxima a 100%, e não foi afetada pela quantidade de nós egoístas no sistema.

Apenas quando comparado a taxa de autenticação dos nós é que o SG-PKM teve um desempenho menor que o PGP-*Like*. Nesse caso, independente da quantidade de nós egoístas, o PGP-*Like* sempre consegue autenticar 100% dos nós do sistema. Já o SG-PKM, embora também não seja afetado pela não-cooperação dos nós egoístas, consegue obter uma taxa de autenticação dos nós, no melhor caso, de 70%. Isso ocorre também devido à necessidade da formação de no mínimo duas cadeias de certificados de grupos para autenticar os nós. Essa restrição, que torna o sistema robusto a ataques *Sybil*, incorpora um custo ao sistema: a diminuição da taxa de autenticabilidade dos nós.

Quando avaliado em cenários com ataques *Sybil*, o SG-PKM obteve um desempenho muito melhor que o PGP-*Like*. Como apresentado, o PGP-*Like* é totalmente vulnerável aos ataques *Sybil*, mesmo na presença de apenas 5% de nós maliciosos. Já o SG-PKM conseguiu resistir a esses ataques: na maioria dos casos, as identidades falsas não consegui-

ram comprometer a quantidade necessária de nós para participarem dos grupos e, mesmo quando conseguiram participar, na maioria das vezes não foram autenticadas pelos nós não-comprometidos. Essas resistência aconteceu, principalmente, devido à necessidade da formação de no mínimo duas cadeias de certificados de grupos para autenticar os nós. Concluindo, o SG-PKM apresenta uma boa robustez contra ataques de falta de cooperação e *Sybil*, resistindo melhor a esses ataques do que o PGP-*Like*.

Os trabalhos futuros incluem: avaliar a eficácia SG-PKM em cenários com outros tipos de ataques comuns nas MANETs; avaliar a praticabilidade de SG-PKM considerando outras características das redes sociais; analisar o impacto no desempenho e na eficácia contra ataques se forem utilizadas mais cadeias disjuntas de certificados na autenticação; verificar o impacto da validação proativa dos certificados emitidos durante a fase de troca de certificados; avaliar o SG-PKM diante de outros tipos de ataques e considerando características como a restrição de energia; realizar simulações utilizando grafos de redes sociais verdadeiras.

REFERÊNCIAS

[1] Keyanalyze - analysis of a large OpenPGP ring, 2009. Acessado em março de 2009.

[2] Atul Adya, William J. Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R. Douceur, Jon Howell, Jacob R. Lorch, Marvin Theimer, e Roger P. Wattenhofer. FARSITE: Federated, available, and reliable storage for an incompletely trusted environment. *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, páginas 1–14, New York, NY, USA, 2002. ACM.

[3] Farooq Anjum e Petros Mouchtaris. *Security for wireless ad hoc networks*. John Wiley & Sons, Hoboken, New Jersey, 2007.

[4] Albert-Laszlo Barabási e Reka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, Oct de 1999.

[5] Albert-Laszlo Barabási, Reka Albert, e Hawoong Jeong. Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications*, 272:173–187, Jul de 1999.

[6] Immanuel M. Bomze, Marco Budinich, Panos M. Pardalos, e Marcello Pelillo. *The Maximum Clique Problem*, capítulo 1, páginas 1–74. Kluwer Academic Publishers, Heidelberg, Germany, 1999.

[7] Levente Buttyán e Jean-Pierre Hubaux. Report on a working session on security in wireless ad hoc networks. *Mobile Computing Communications Review (SIGMOBILE)*, 7(1):74–94, 2003.

[8] Srdjan Čapkun, Levente Buttyán, e Jean-Pierre Hubaux. Small worlds in security systems: an analysis of the PGP certificate graph. *Proceedings of the 2002 Workshop on New Security Paradigms (NSPW '02)*, páginas 28–35, New York, NY, USA, 2002. ACM.

[9] Srdjan Čapkun, Levente Buttyán, e Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, 2003.

[10] Srdjan Čapkun, Jean-Pierre Hubaux, e Levente Buttyán. Mobility helps security in ad hoc networks. *Proceedings of the 4th ACM International Symposium on Mobile Ad hoc Networking & Computing (MobiHoc '03)*, páginas 46–56, New York, NY, USA, 2003. ACM.

[11] Srdjan Čapkun, Jean-Pierre Hubaux, e Levente Buttyán. Mobility helps peer-to-peer security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, 2006.

[12] Haowen Chan e Adrian Perrig. PIKE: Peer intermediaries for key establishment in sensor networks. *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, volume 1, páginas 524–535, Washington, DC, USA, 2005. IEEE Computer Society.

[13] Haowen Chan, Adrian Perrig, e Dawn Song. Random key predistribution schemes for sensor networks. *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03)*, páginas 197–213, Washington, DC, USA, 2003. IEEE Computer Society.

[14] Ting-Yi Chang, Chou-Chen Yang, e Min-Shiang Hwang. A threshold signature scheme for group communications without a shared distribution center. *Future Generation Computer System*, 20(6):1013–1021, 2004.

[15] Alice Cheng e Eric Friedman. Sybilproof reputation mechanisms. *Proceedings of the 3rd Workshop on Economics of Peer-to-Peer Systems (P2P-Econ '05)*, páginas 128–132, New York, NY, USA, 2005. ACM.

[16] Imrich Chlamtac, Marco Conti, e Jennifer J.-N. Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, 2003.

[17] Bruce Christianson e William S. Harbison. Why isn't trust transitive? *Proceedings of the 6th International Workshop on Security Protocols*, páginas 171–176, London, UK, 1997. Springer-Verlag.

[18] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, e W. Polk. *RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Marina del Rey, CA, USA, May de 2008.

[19] L. F. Costa, F. A. Rodrigues, G. Travieso, e P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances In Physics*, 56:167–242, 2007.

[20] Eduardo da Silva, Michele N. Lima, Aldri L. dos Santos, e Luiz C. P. Albini. Quantifying misbehaviour attacks against the self-organized public key management on manets. *Proceedings of the International Conference on Security and Cryptography (SECRYPT '08)*, páginas 128–135, Porto, Portugal, Jul de 2008. INSTCC Press.

[21] George Danezis e Prateek Mittal. SybilInfer: Detecting sybil nodes using social networks. *Proceedings of the 16th Annual Network & Distributed System Security Symposium (NDSS '09)*, Reston, VA, USA, Feb de 2009. Internet Society.

[22] Sanjay K. Dhurandher e G. V. Singh. Weight based adaptive clustering in wireless ad hoc networks. *Proceedings of the IEEE International Conference on Personal Wireless Communications (ICPWC '05)*, páginas 95–100, Washington, DC, USA, 2005. IEEE Computer Society.

[23] Whitfield Diffie e Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[24] Djamel Djenouri, Lyes Khelladi, e Ndjib Badache. A survey of security issues in mobile ad hoc and sensor networks. *IEEE Surveys and Tutorials*, 7(4):2–28, 2005.

[25] John R. Douceur. The Sybil attack. *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '01)*, páginas 251–260, London, UK, 2001. Springer-Verlag.

[26] Wenliang Du, Jing Deng, Yunghsiang S. Han, Shigang Chen, e Pramod K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, Washington, DC, USA, Mar de 2004. IEEE Computer Society.

[27] Wenliang Du, Jing Deng, Yunghsiang S. Han, e Pramod K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, páginas 42–51, New York, NY, USA, 2003. ACM.

[28] P. Erdös. On cliques in graphs. *Israel Journal of Mathematics*, 4:233–234, dec de 1996.

[29] Laurent Eschenauer e Virgil D. Gligor. A key management scheme for distributed sensor networks. *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, páginas 41–47, New York, NY, USA, 2002. ACM Press.

[30] L.M. Feeney, B. Ahlgren, e A. Westerlund. Spontaneous networking: an application oriented approach to ad hoc networking. *IEEE Communications Magazine*, 39(6):176–181, Jun de 2001.

[31] Hossein Ghodosi e Rei Safavi-naini. Dynamic threshold cryptosystems: A new scheme in group oriented cryptography. *Proceedings of the International Conference on the 1st Theory and Applications of Cryptology (PRAGOCRYPT '96)*, páginas 370–379, Prague, Czech, 1996. Czech Technical University Publishing House.

[32] Mohamed G. Gouda e Eunjin Jung. Certificate dispersal in ad-hoc networks. *Proceedings of the 24th International Conference on Distributed Computing Systems*

(ICDCS '04), páginas 616–623, Washington, DC, USA, 2004. IEEE Computer Society.

[33] Wenbo He, Ying Huang, Klara Nahrstedt, e Whay C. Lee. SMOCK: A self-contained public key management scheme for mission-critical wireless ad hoc networks. *Proceedings of the 5th IEEE International Conference on Pervasive Computing and Communications (PERCOM '07)*, páginas 201–210, Washington, DC, USA, 2007. IEEE Computer Society.

[34] Anne Marie Hegland, Eli Winjum, Stig F. Mjolsnes, Chunmig Rong, Oivind Kure, e Pal Spilling. A survey of key management in ad hoc networks. *IEEE Communications Surveys*, 08(03):48–66, 2006.

[35] Jean-Pierre Hubaux, Levente Buttyán, e Srdan Čapkun. The quest for security in mobile ad hoc networks. *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '01)*, páginas 146–155, New York, NY, USA, 2001. ACM.

[36] IEEE. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, NY, 1997. IEEE Strandard 802.11-1997.

[37] Aram Khalili, Jonathan Katz, e William A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT '03)*, páginas 342, Washington, DC, USA, 2003. IEEE Computer Society.

[38] Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, e Lixia Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. *Proceedings of the 9th International Conference on Network Protocols (ICNP '01)*, páginas 251, Washington, DC, USA, 2001. IEEE Computer Society.

[39] M. Latapy, C. Magnien, e N. Del Vecchio. Basic notions for the analysis of large two-mode networks. *Social Networks*, 30(1):31–48, 2008.

[40] Michele N. Lima, Helber W. da Silva, Aldri L. dos Santos, e Guy Pujolle. An architecture for survivable mesh networking. *Proceedings of the 2008 IEEE Global Communications Conference (GLOBECOM '08)*, páginas 688–692, Los Alamitos, CA, USA, 2008. IEEE Communications Society.

[41] Michele Nogueira Lima, Aldri Luiz dos Santos, e Guy Pujolle. A survey of survivability in mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 11:66–77, Feb de 2009.

[42] Haiyun Luo, Jiejun Kong, Petros Zerfos, Songwu Lu, e Lixia Zhang. URSA: ubiquitous and robust access control for mobile ad hoc networks. *IEEE/ACM Transactions on Networking*, 12(6):1049–1063, 2004.

[43] A. J. Menezes, P. C. V. Oorschot, e S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Danvers, MA, USA, 1996.

[44] Pietro Michiardi e Refik Molva. Ad hoc networks security. *ST Journal of System Research*, 4(1), Mar de 2003.

[45] Revik Molva e Pietro Michiardi. Security in ad hoc networks. *Proceeding of 8th IFIP International Conference on Personal Wireless Communications (PWC '03)*, Venice, Italy, Sep de 2003. IFIP. Also published as LNCS Volume 2775.

[46] James Newsome, Elaine Shi, Dawn Song, e Adrian Perrig. The Sybil attack in sensor networks: analysis & defenses. *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, páginas 259–268, New York, NY, USA, 2004. ACM.

[47] Edith C. H. Ngai e Michael R. Lyu. Trust- and clustering-based authentication services in mobile ad hoc networks. *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)*, páginas 582–587, Washington, DC, USA, 2004. IEEE Computer Society.

[48] NS-2. The network simulator - ns-2, 2008. Acessado em março de 2009.

[49] Panagiotis Papadimitratos e Zygmunt J. Haas. *Securing mobile ad hoc networks*, capítulo 21, páginas 457–481. CRC Press - Auerbach Publications, Boca Raton, Florida, USA, 2005.

[50] Torben Pryds Pedersen. A threshold cryptosystem without a trusted party (extended abstract). *Proceedings of Advances in Cryptology (EuroCrypt '91)*, volume 547 of *Lecture Notes in Computer Science*, páginas 522–526, London, UK, 1991. Springer.

[51] Chris Piro, Clay Shields, e Brian Neil Levine. Detecting the Sybil attack in ad hoc networks. *Proceeding of the IEEE/ACM International Conference on Security and Privacy in Communication Networks (SecureComm '06)*, páginas 1–11, New York, NY, USA, August de 2006. ACM.

[52] Theodore S. Rappaport. *Comunicação Sem Fio: Princípios E Práticas*. Pearson, São Paulo, SP, Brasil, 2 edition, dez de 2008.

[53] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, e Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[54] Nitesh Saxena, Gene Tsudik, e Jeong Hyun Yi. Access control in ad hoc groups. *Proceedings of the 1st International Workshop on Hot Topics in Peer-to-Peer Systems (HOT-P2P '04)*, páginas 2–7, Washington, DC, USA, 2004. IEEE Computer Society.

[55] Vivek Shah, Hongmei Deng, e Dharma P. Agrawal. Parallel cluster formation for secured communication in wireless ad hoc networks. *Proceedings of the 12th IEEE International Conference on Networks (ICON '04)*, volume 2, páginas 475–479, Washington, DC, USA, nov de 2004. IEEE Computer Society.

[56] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[57] Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949.

[58] R. Shirey. *RFC 2828: Internet security glossary.* Marina del Rey, CA, USA, May de 2000.

[59] Willian Stallings. *Criptografia e Segurança de Redes: Princípios e Práticas.* Prentice Hall, São Paulo, SP, Brasil, 4 edition, 2009.

[60] Nguyen Tran, Bonan Min, Jinyang Li, e Lakshminarayanan Subramanian. Sybil-resilient online content voting. *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI '09)*, páginas 15–28, Boston, MA, USA, Apr de 2009. USENIX Association.

[61] S. Tsukiyama, M. Ide, H. Ariyoshi, e I. Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.

[62] Johann van der Merwe, Dawoud Dawoud, e Stephen McDonald. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computing Survey*, 39(1):1, 2007.

[63] Fabien Viger e Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. *Proceedings of 11th Annual International Conference of Computing and Combinatorics (COCOON '05)*, volume 3595 of *LNCS*, páginas 440–449, London, UK, 2005. Springer.

[64] Shiuh-Jeng Wang, Yuh-Ren Tsai, e Chung-Wei Chen. Strategies averting Sybil-type attacks based on the Blom-scheme in ad hoc sensor networks. *Journal of Communications (JCM)*, 3(1):20–26, 2008.

[65] Xiao F. Wang e Guanrong Chen. Complex networks: small-world, scale-free and beyond. *IEEE Circuits and Systems Magazine*, 3(1):6–20, 2003.

[66] D. J. Watts e S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun de 1998.

[67] Bing Wu, Jianmin Chen, Jie Wu, e Mihaela Cardei. *A survey on attacks and countermeasures in mobile ad hoc networks*, capítulo 12, páginas 103–136. Springer-Verlag, New York, NY, USA, 2006.

[68] Bing Wu, Jie Wu, Eduardo B. Fernandez, Mohammad Ilyas, e Spyros Magliveras. Secure and efficient key management in mobile ad hoc networks. *Journal of Network and Computer Applications*, 30(3):937–954, 2005.

[69] J. Wu e D. J. Watts. Small worlds: the dynamics of networks between order and randomness. *ACM SIGMOD Record*, 31(4):74–75, 2002.

[70] Seung Yi e Robin Kravets. MOCA: Mobile certificate authority for wireless ad hoc networks. *Proceedings of the 2nd Annual PKI Research Workshop (PKI '03)*, Gaithersburg, MD, USA, 2003. NIST – National Institute of Standards and Technology.

[71] Seung Yi e Robin Kravets. Composite key management for ad hoc networks. *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous' 04)*, páginas 52–61, New York, NY, USA, Aug de 2004. ACM.

[72] Chi Zhang, Yang Song, e Yuguang Fang. Modeling secure connectivity of self-organized wireless ad hoc networks. *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, Los Alamitos, CA, USA, 2008. IEEE Communications Society.

[73] Lidong Zhou e Zygmunt J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

[74] Philip R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, MA, USA, 1995.

APÊNDICE A

PGP-LIKE DIANTE DE ATAQUES FALTA DE COOPERAÇÃO EM CENÁRIOS DE 1500 X 300 METROS

Este apêndice apresenta os resultados da avaliação do impacto dos ataques de falta de cooperação, em cenário com tamanho de 1500 x 300 metros, no PGP-*Like*. Os resultados apresentados para esses cenários são similares aos resultados para cenários com tamanho de 1000 x 1000 metros, apresentados na Seção 2.3.1, porém é necessário um tempo menor para a convergência das trocas de certificados e da alcançabilidade dos nos.

Convergência das trocas de certificados

A Figura A.1 apresenta a convergência das trocas de certificados (CE) em cenários com o raio de alcance das antenas igual a 120 metros. Como nos casos com o tamanho do ambiente igual a 1000 x 1000 metros, o aumento de atacantes resulta em uma queda em CE. Em cenários com 40% de nós egoístas, o valor de CE chega no máximo a 83%. Porém, na presença de até 20% de nós egoístas, o valor de CE é pouco afetado.

Se comparado com os resultados em cenários com tamanho igual a 1000 x 1000 metros, é possível notar que o tamanho do ambiente não afeta o resultado final dos valores de CE em cenários com ou sem ataque. No caso dos resultados em cenários de tamanho 1500 x 300 metros, apenas é necessário um tempo menor para que CE alcance o seu valor máximo. Em cenários com 40% de nós egoístas, o valor de CE estabilizou logo após os primeiros 1000 segundos de vida da rede.

É possível notar que com o aumento da velocidade de movimentação dos nós, é necessário um tempo menor para a convergência das trocas de certificados. Em cenários sem ataques e com velocidade de movimentação igual a 20 m/s, o tempo de convegência ocorre aproximadamente após 400 segundos de vida da rede. Já com uma velocidade de movimentação igual a 5 m/s, esse tempo é de 600 segundos, cerca de 50% maior.

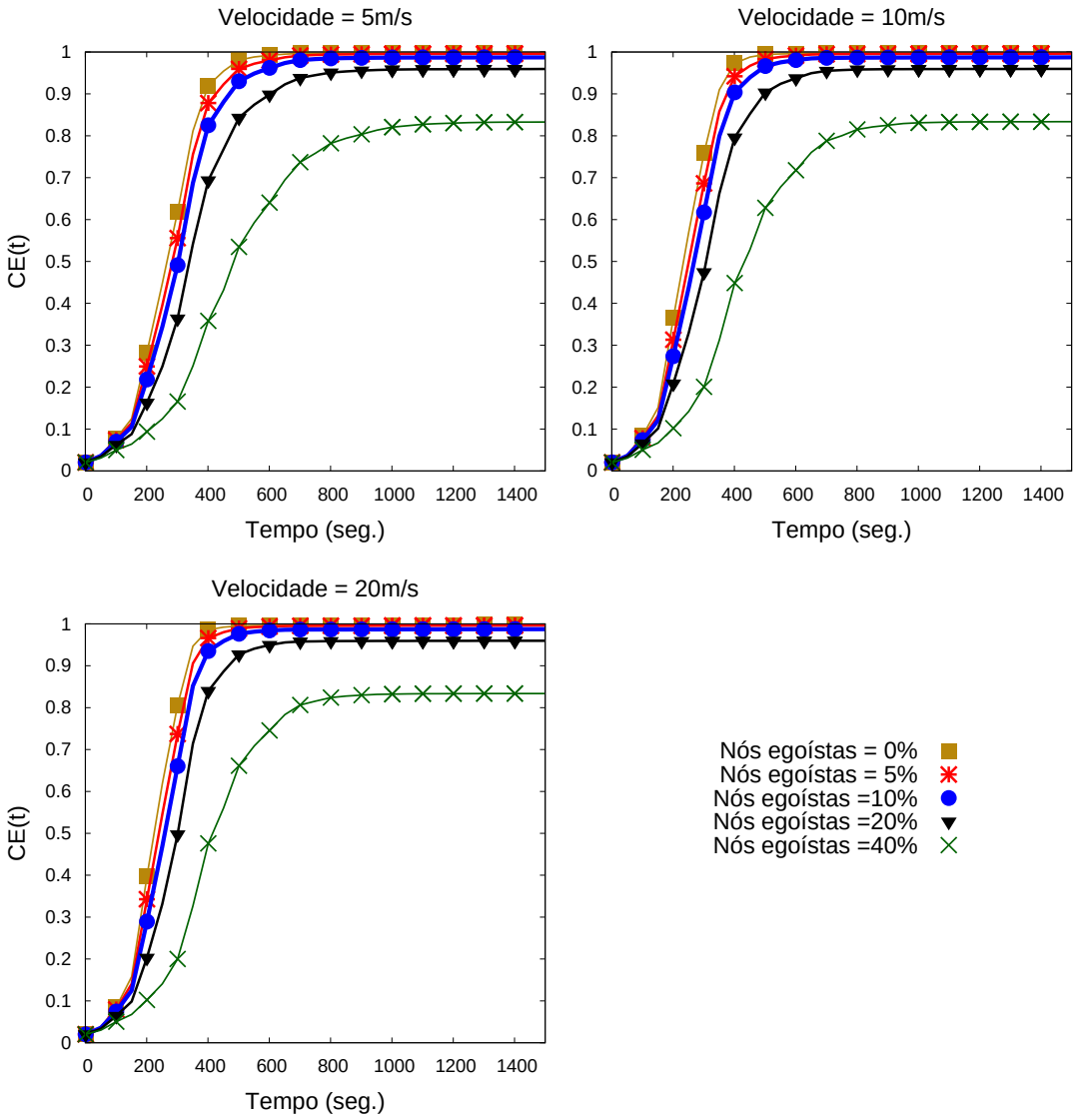


Figura A.1: Convergência das trocas de certificados diante de ataques de falta de cooperação (1500 x 300 metros e raio de 120 metros)

A Figura A.2 apresenta os valores de CE em cenários com o raio de alcance das antenas igual a 50 metros. Nesse último caso, o impacto dos ataques sobre o valor de CE é o mesmo dos cenários anteriores. Como no cenário com o tamanho da rede igual a 1000 x 1000 metros e o raio de comunicação dos nós igual a 50 metros (Seção 2.3.1), é necessário um tempo maior para a estabilização do valor de CE, se comparado com cenários em que o raio de alcance é de 120 metros.

Entretanto, independente da mudança do raio de alcance das antenas, o impacto do ataque de falta de cooperação, após a convergência das trocas de certificados, é o mesmo. Nesse caso, em cenários com até 20% de nós egoístas, o valor de CE quase não é afetado. Já para uma quantidade de 40% de nós maliciosos, CE alcança um valor máximo de 83%.

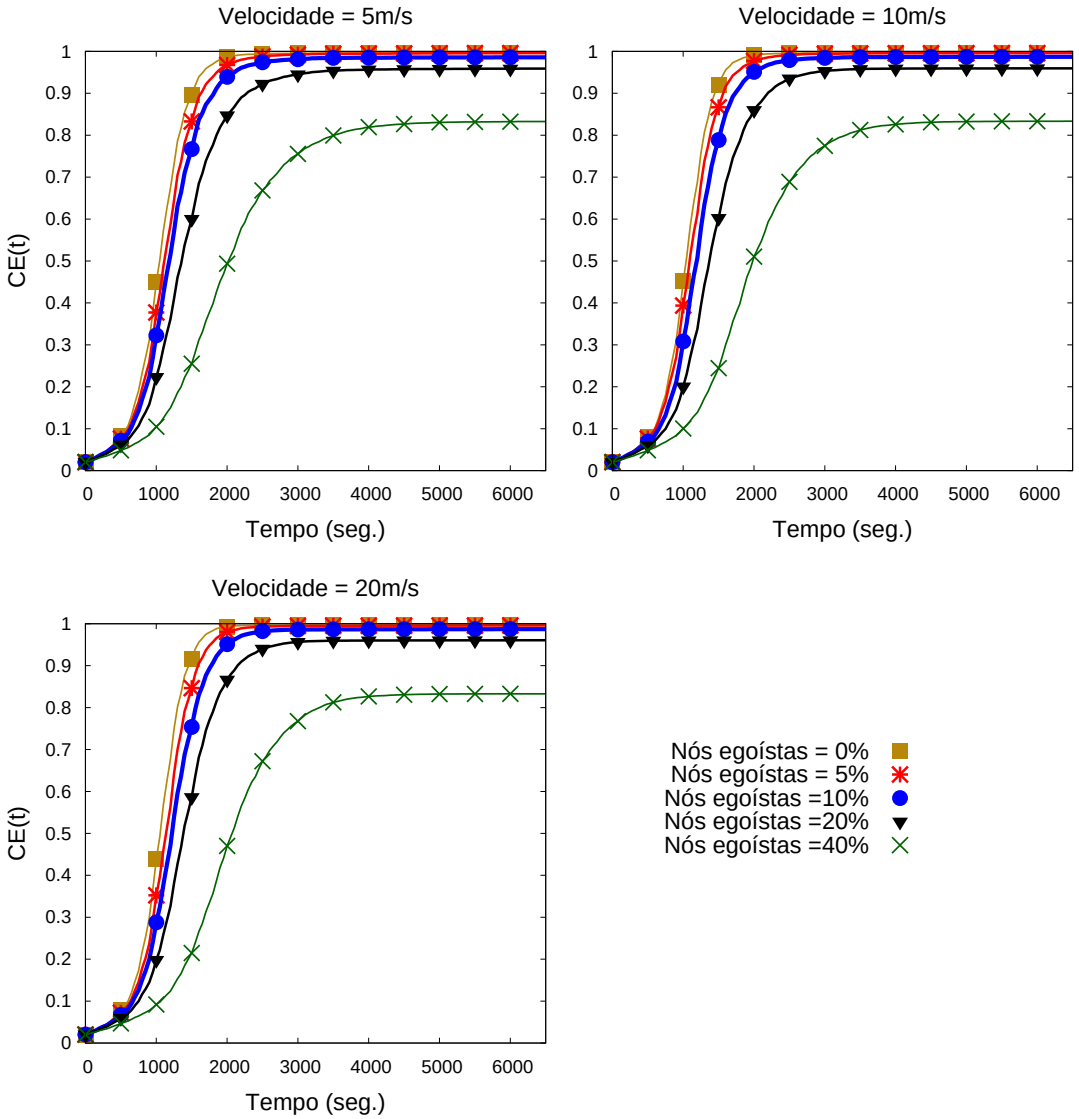


Figura A.2: Convergência das trocas de certificados diante de ataques de falta de cooperação (1500 x 300 metros e raio de 50 metros)

Alcançabilidade dos nós

As Figuras A.3 e A.4 apresentam o valor de UR em cenários com o raio de alcance das antenas igual a 120 e 50 metros, respectivamente. Nos dois casos nota-se que, mesmo na presença de até 40% de nós egoístas, o valor de UR não é afetado. Apenas nos casos com o raio de alcance igual a 50 metros, é necessário um tempo maior para que a alcançabilidade dos nós alcance 100%.

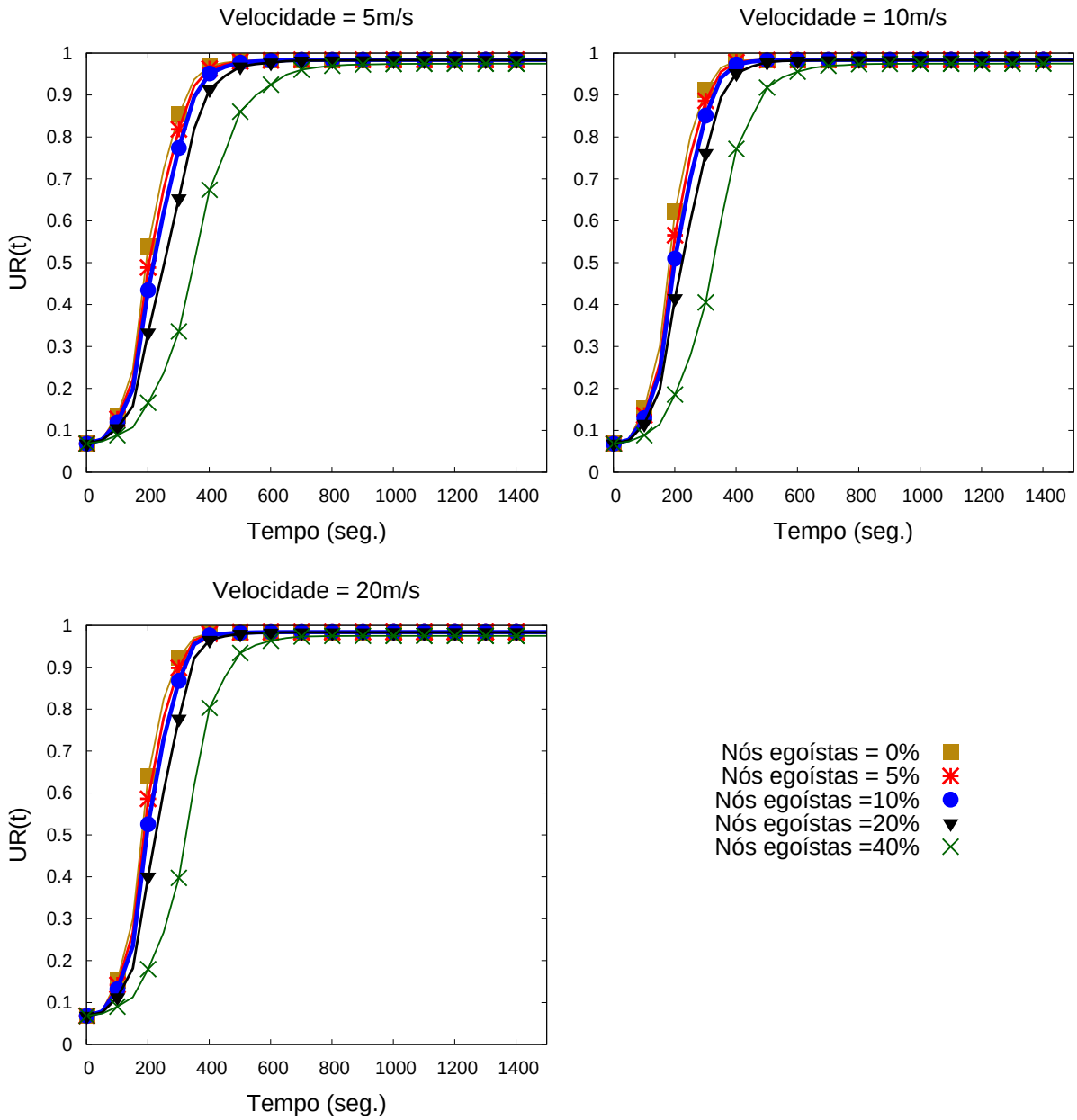


Figura A.3: Alcançabilidade dos nós diante de ataques de falta de cooperação (1500 x 300 metros e raio de 120 metros)

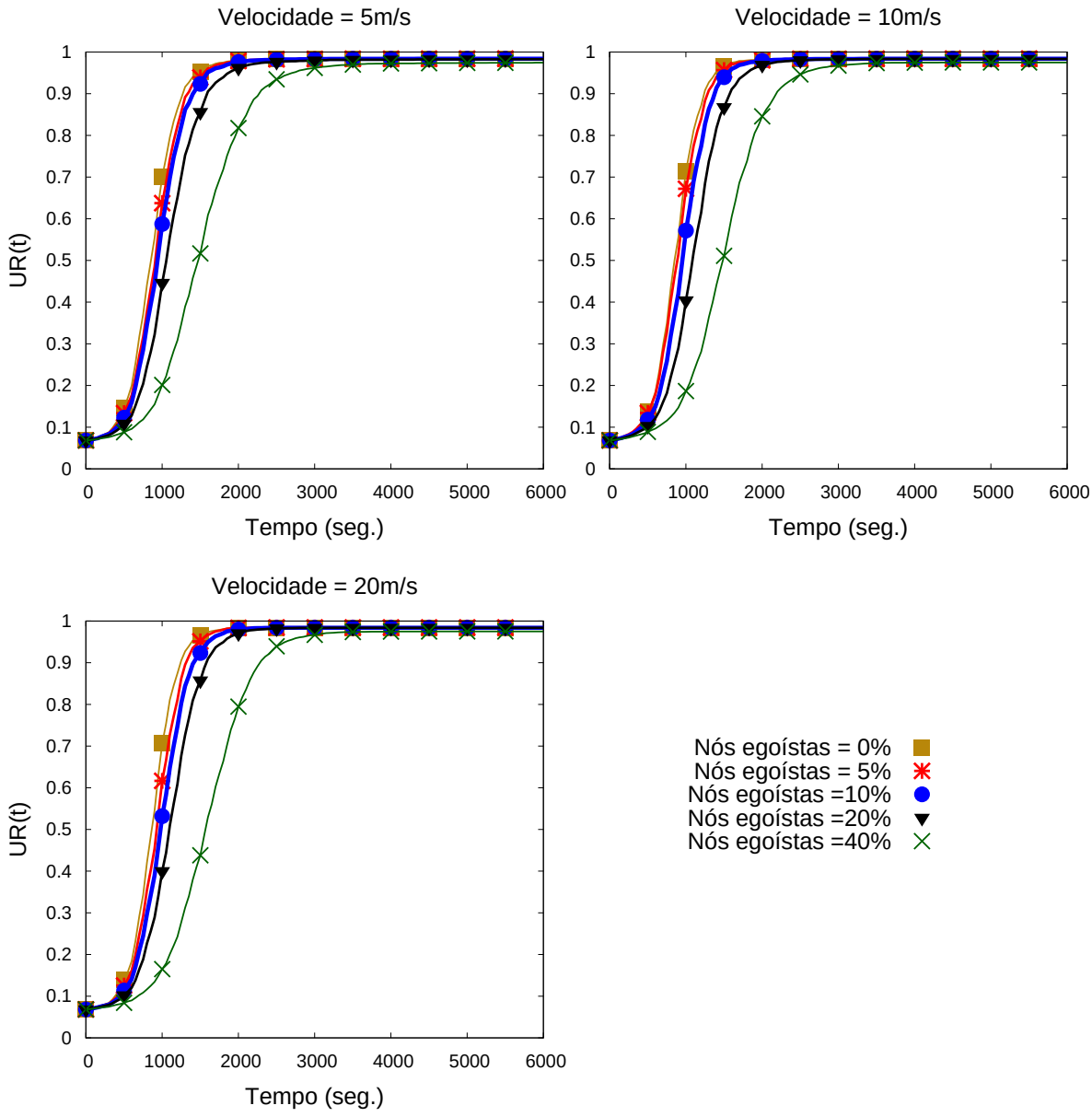


Figura A.4: Alcançabilidade dos nós diante de ataques de falta de cooperação (1500 x 300 metros e raio de 50 metros)

APÊNDICE B

SG-PKM DIANTE DE ATAQUES FALTA DE COOPERAÇÃO EM CENÁRIOS DE 1500 X 300 METROS

Este apêndice apresenta os resultados da avaliação do impacto dos ataques de falta de cooperação, em cenário com tamanho de 1500 x 300 metros, na métrica CE do SG-PKM. Os resultados apresentados para esses cenários são similares aos resultados para cenários com tamanho de 1000 x 1000 metros, apresentados na Seção 4.3.3, porém é necessário um tempo menor para a convergência das trocas de certificados e da alcançabilidade dos nos. Os parâmetros utilizados nesses simulações são os mesmos apresentados na Seção 4.3.2.

A Figura B.1 apresenta a convergência das trocas de certificados (CE) em cenários com o raio de alcance das antenas igual a 120 metros e velocidades de 5 m/s. Como nos casos com o tamanho do ambiente igual a 1000 x 1000 metros, em todos os casos, o valor de CE sempre alcança 100%, independente da quantidade de atacantes. Se comparado com o PGP-*Like*, no SG-PKM é necessário um tempo menos para a convergência do sistema.

A Figura B.2 apresenta a convergência das trocas de certificados (CE) em cenários com o raio de alcance das antenas igual a 120 metros e velocidades de 20 m/s. Da mesma forma, em todos os casos, o valor de CE sempre alcança 100%, independente da quantidade de atacantes.

Por fim, a Figura B.3 apresenta a convergência das trocas de certificados (CE) em cenários com o raio de alcance das antenas igual a 50 metros e velocidades de 20 m/s. Como nos casos anteriores, independente da quantidade de nós egoístas no sistema, o valor de CE sempre alcança 100%.

Se comparado com os resultados em cenários com tamanho igual a 1000 x 1000 metros, é possível notar que o tamanho do ambiente não afeta o resultado final dos valores de CE em cenários com ou sem ataque. No caso dos resultados em cenários de tamanho

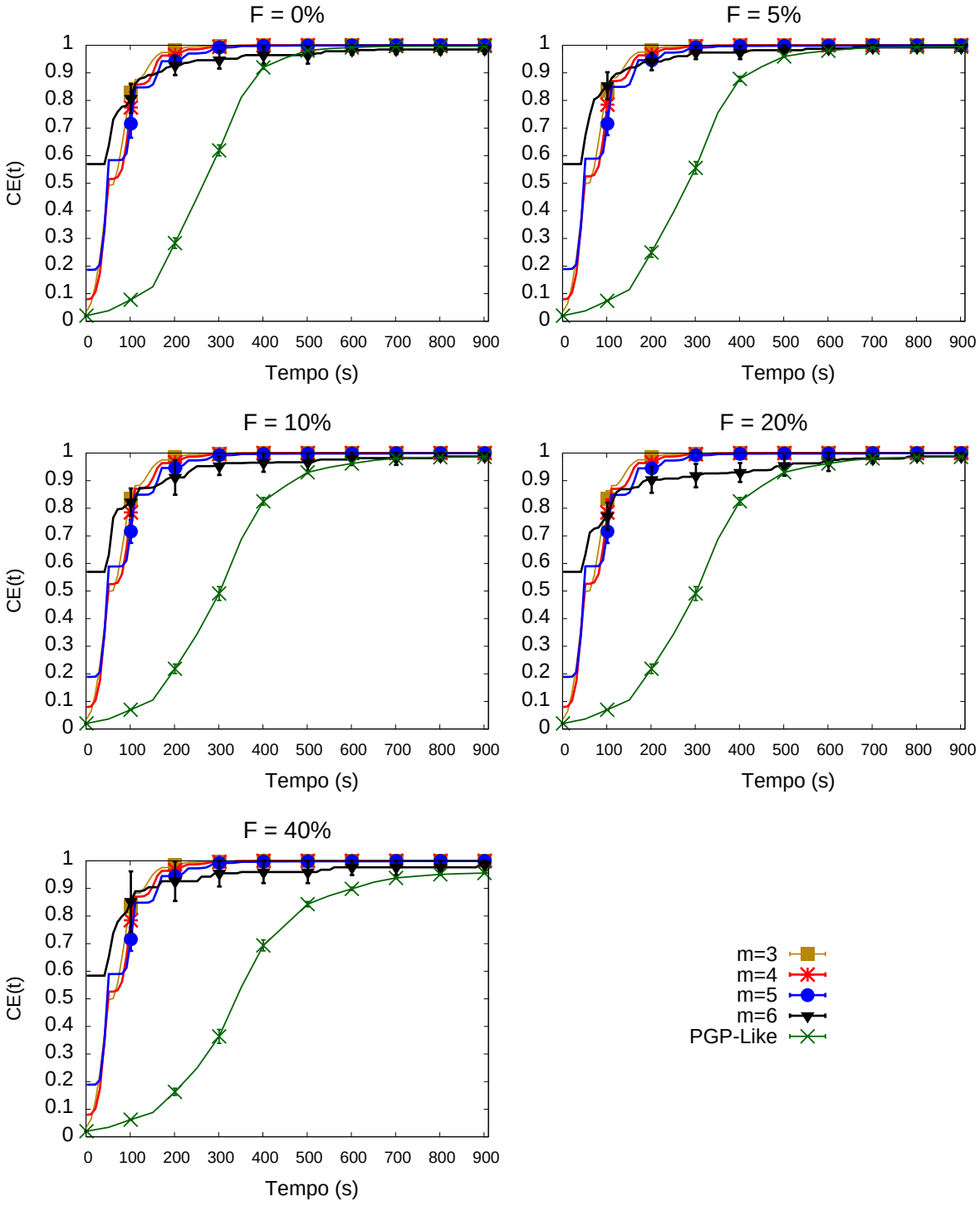


Figura B.1: Tempo de convergência com velocidade de 5 m/s e raio de 120 metros

1500 x 300 metros, apenas é necessário um tempo menor para que CE alcance o seu valor máximo.

É possível notar, também, que com o aumento da velocidade de movimentação dos nós, é necessário um tempo menor para a convergência das trocas de certificados. Em

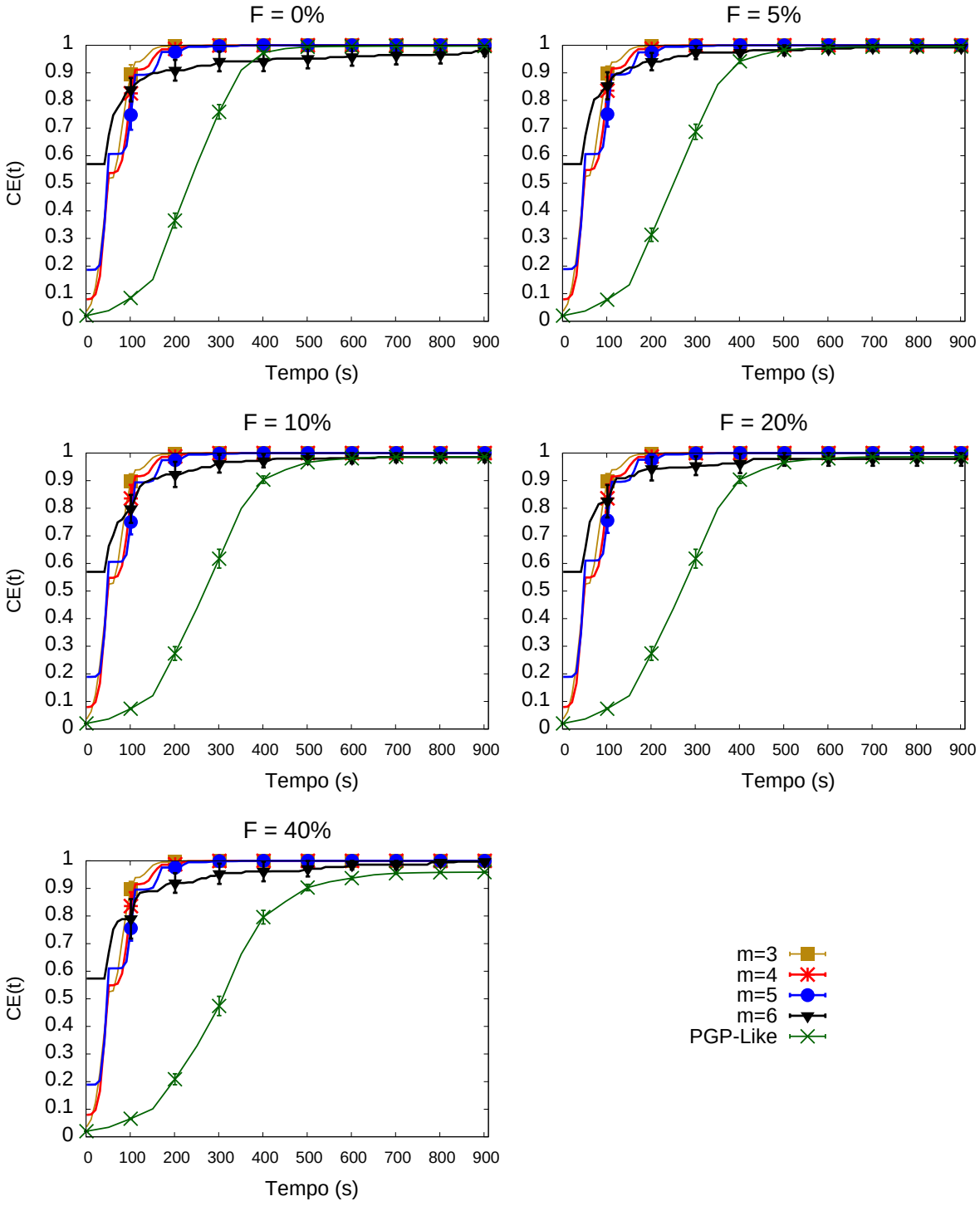


Figura B.2: Tempo de convergência com velocidade de 10 m/s e raio de 120 metros

cenários sem ataques, com velocidade de movimentação igual a 5 m/s (Figura B.1), e tamanhos de grupo igual a 4 ou 5, a convergência do sistema acontece aproximadamente após 300 segundos de vida da rede. Já nos cenário com velocidade igual a 10 m/s, esse tempo é de apenas 200 segundos.

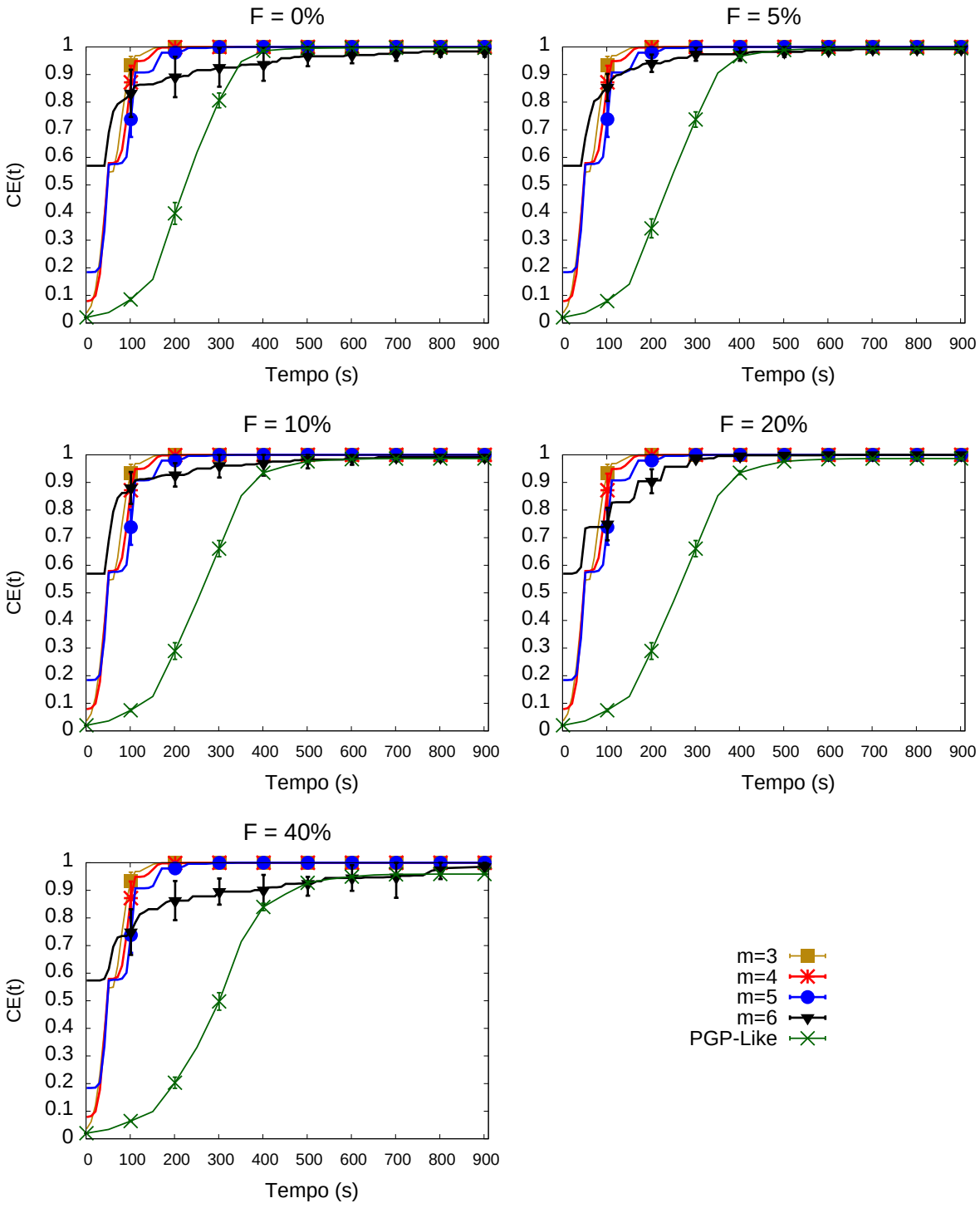


Figura B.3: Tempo de convergência com velocidade de 20 m/s e raio de 120 metros

As figuras B.4, B.5 e B.6 apresentam os resultados nas simulações em cenários com raio de alcance igual a 50 metros e velocidades de movimentação igual a 5 m/s, 10 m/s e 20 m/s, respectivamente. Em todos os casos, como nos cenários com raio de alcance igual a 120

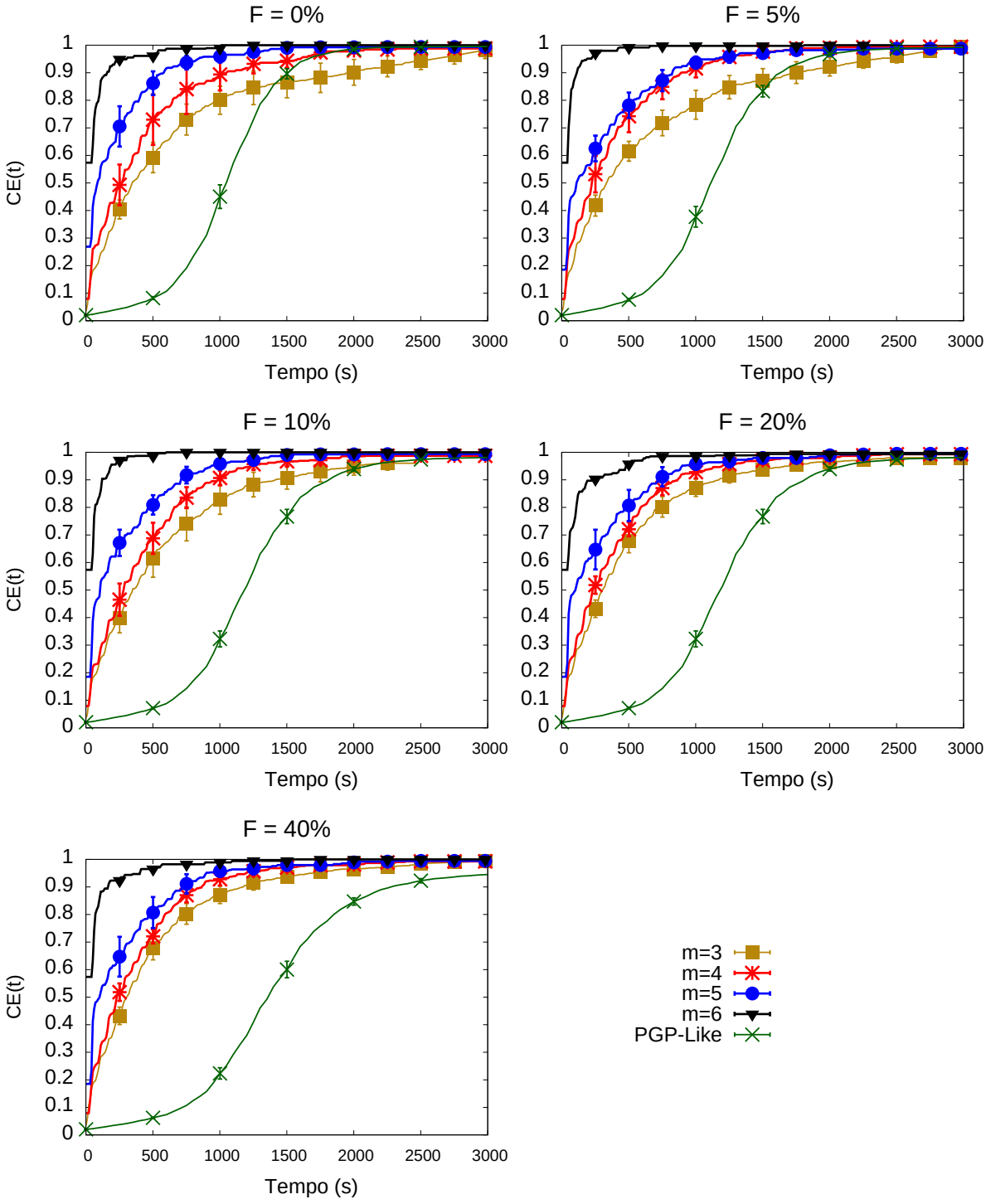


Figura B.4: Tempo de convergência com velocidade de 5 m/s e raio de 50 metros

metros, CE sempre alcança 100%, independente de quantidade de nós egoístas no sistema. Esses resultados mostram que o sistema possui uma boa sobrevivência aos ataques de falta de cooperação, independente dos cenários que são utilizados e da densidade da rede.

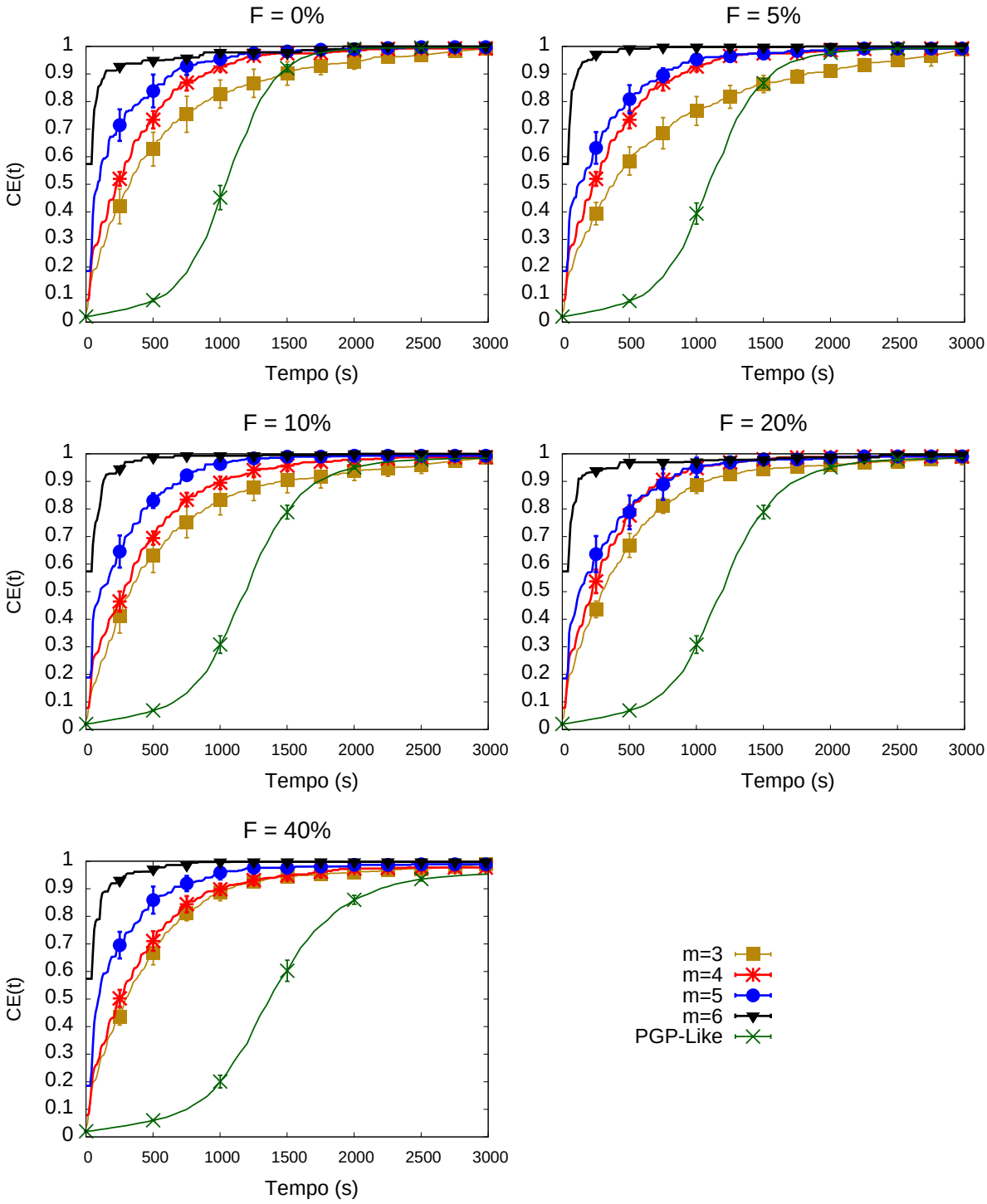


Figura B.5: Tempo de convergência com velocidade de 10 m/s e raio de 50 metros

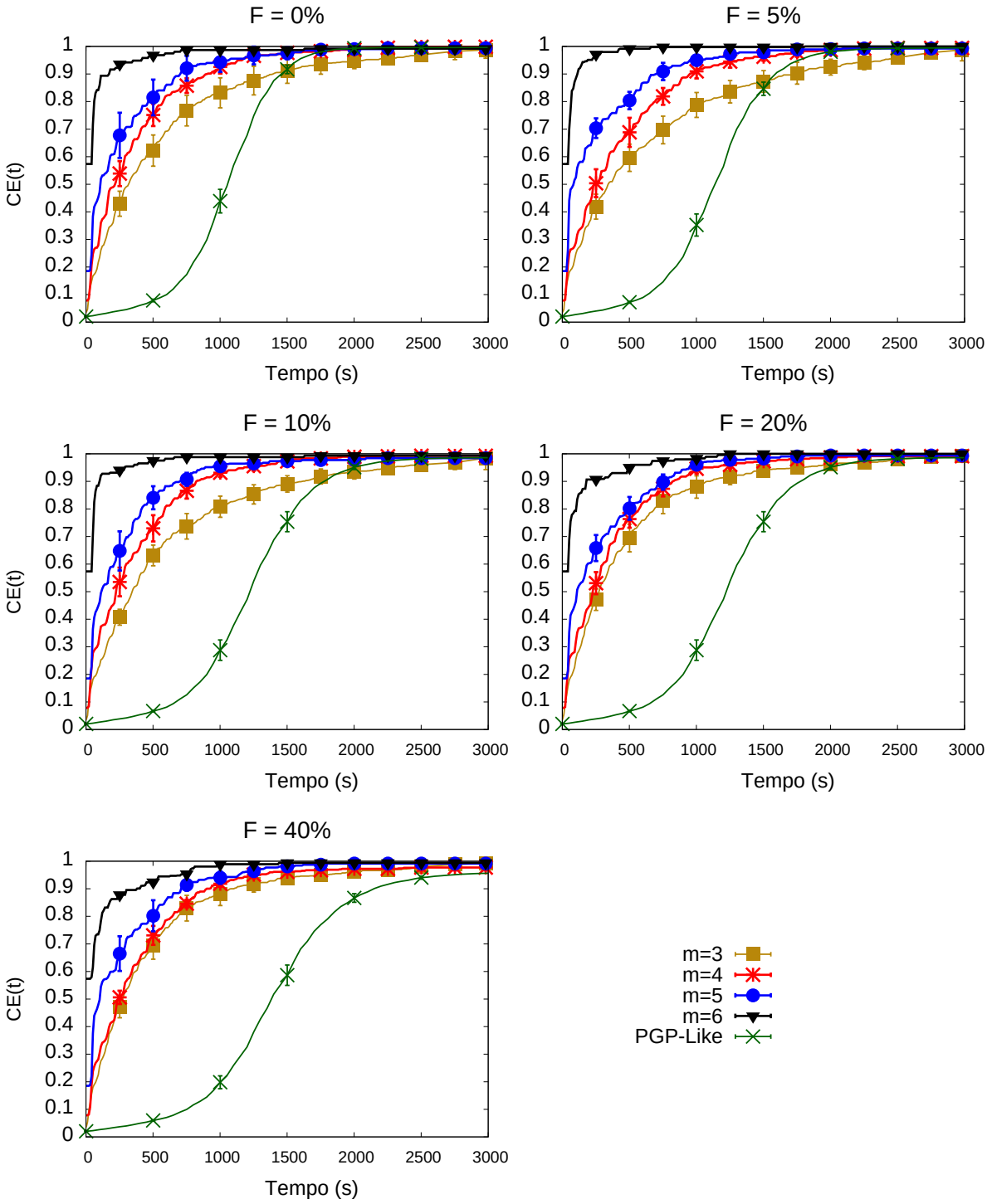


Figura B.6: Tempo de convergência com velocidade de 20 m/s e raio de 50 metros

APÊNDICE C

LISTA DE PUBLICAÇÕES

Os estudos realizados para este trabalho resultaram em algumas publicações, que estão listadas abaixo:

1. Eduardo da Silva, Michele N. Lima, Aldri L. Santos e Luiz Carlos P. Albini. Identity-based key management in mobile ad hoc networks: Techniques and applications. *IEEE Wireless Communications Magazine*, IEEE Communications Society, New York, NY, USA, v. 15, Oct 2008. ISSN 1536-1284.

2. Eduardo da Silva, Michele N. Lima, Aldri L. Santos e Luiz Carlos P. Albini. Quantifying misbehaviour attacks against the self-organized public key management on manets. In: *Proceedings of the International Conference on Security and Cryptography (SECRYPT '08)*. Porto, Portugal: INSTCC Press, 2008. p. 128–135. ISSN 978-989-8111-59-3. **Na lista dos best-papers da conferência**

3. Angelo Bannack, Eduardo da Silva, Michele N. Lima, Aldri L. Santos e Luiz Carlos P. Albini. Segurança em redes ad hoc. In: *Anais do XXVI Simpósio Brasileiro de Telecomunicações (SBRT '08)*. Rio de Janeiro, RJ, Brasil: SBrT - Sociedade Brasileira de Telecomunicaçõs, 2008. p. 19–20. ISBN 978-85-89748-05-6.

4. Michele N. Lima, Eduardo da Silva, Luiz Carlos P. Albini, Aldri L. Santos e Guy Pujolle. Survivable Keying for Wireless Ad Hoc Networks. In: *11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009)*, New York, June, 2009. p. 606-613. ISSN 978-1-4244-3487-9.

5. Renan Fischer e Silva, Eduardo da Silva e Luiz Carlos P. Albini. Resisting Impersonation Attacks in Chaining-Based Public-Key Management on MANETs: the Virtual Public-Key Management. In: *Proceedings of the International Conference*

on Security and Cryptography (SECRYPT '09). Milan, Italy: INSTCC Press, 2009. (to appear)

6. Eduardo da Silva, Michele N. Lima, Aldri L. Santos e Luiz Carlos P. Albini. Chapter: Analyzing the Effectiveness of Self-Organized Public Key Management on MANETs under Lack of Cooperation and Impersonation attacks. In: *E*-Business and Telecommunication Networks. (Springer) 2009 (to appear).